(2)

GE Aerospace
Advanced Technology Laboratories

RASSP Final Technical Report

CLIN 0002AB

Submitted to:
DARPA/ESTO
ATTN: Eliot D. Cohen
3701 North Fairfax Drive
Arlington, VA 22203-1714

Reference:
CMDA972-92-R-0017

Submitted by:
GE Aerospace
Advanced Technology Laboratories
Bldg. 145
Moorestown Corporate Center
Moorestown, New Jersey 08057

Point of Contact:
Technical
J.E. Saultz, (609) 866-6402

October 21, 1992

Table of Contents

List of Figures

List of Tables

St-A per telecon, Mr. Cohen, DARPA/
ESTO. Arlington, VA 22203

JK 11-19-92

1. SUMMARY OF ACCOMPLISHMENTS

1.1 Task Objectives

The overall objective of the DARPA/Tri-Service RASSP program is to "demonstrate a capability to rapidly specify, produce, and yield domain-specific, affordable signal processors for use in Department of Defense systems such as automatic target acquisition, tracking, and recognition, electronic countermeasures, communications, and SIGINT".

The objective of the study phase is to specify a recommended program plan for the government to use as a template for procurement of the RASSP design system and demonstration program. To accomplish that objective, the study phase program tasks are to specify a development methodology for signal processors (adaptable to various organizational design styles, and application areas), analyze the requirements in CAD/CAE tools to support the development methodology, identify the state and development plans of the industry relative to this area, and to recommend the additional developments not currently being addressed by the industry, which are recommended as RASSP developments. In addition, the RASSP study phase will define a linking approach for electronically linking design centers to manufacturing centers so a complete cycle for prototyping can be accomplished with significantly reduced cycle time.

1.2 Technical Problems

Design and implementation systems in use today at major DoD aerospace contractors, and government facilities generally consist of disjoint tool sets, as indicated in Figure 1-1, each of which is focused on addressing a specific aspect or design level of the overall signal processor development process. This is particularly true at higher levels of the design process.

Commercial EDA vendors however have made significant progress in tool development and integration particularly for the lower design levels associated with the digital design and implementation of VLSI chips, printed wiring assemblies and multichip modules. For these frameworks, CAD frameworks are being developed which specifically address integration of multiple tool types into a common environment, sharing of information between tools in a seamless fashion, and providing common support services such as configuration management, methodology management. Extension of these environments to address the coupling of system level design and analysis tools with detailed design, analysis, and manufacturing disciplines is required to eliminate or minimize the inefficiencies in the engineering design processes caused by manual translation of design information between tool sets. Such steps are needed to provide a concurrent engineering capability that spans the design process.

Automatic storage and configuration management of product data and electronic linking of design, manufacturing, and component vendors is just emerging. However, there are many software products that are available to support the desired electronic commerce RASSP will require. The enterprise approach required to support the

RASSP goals has many common requirements for other DoD development areas and, therefore, the RASSP development will provide developments that can be readily adapted to other areas.

Information representation approaches using current hardware descriptive languages, has not been developed adequately to address areas other than digital hardware design. Extensions are required to support higher level system designs, analog designs, and combination hardware/software systems. Further extensions of todays HDLs may not be adequate to address these areas. Approaches which involve several HDLs each of which is both optimized for a particular design area has provisions for linkage or mapping of the information between HDLs should be addressed to achieve the RASSP goals.

Today's design and implementation approaches have significant gaps between system design, design implementation, vendor support and manufacturing

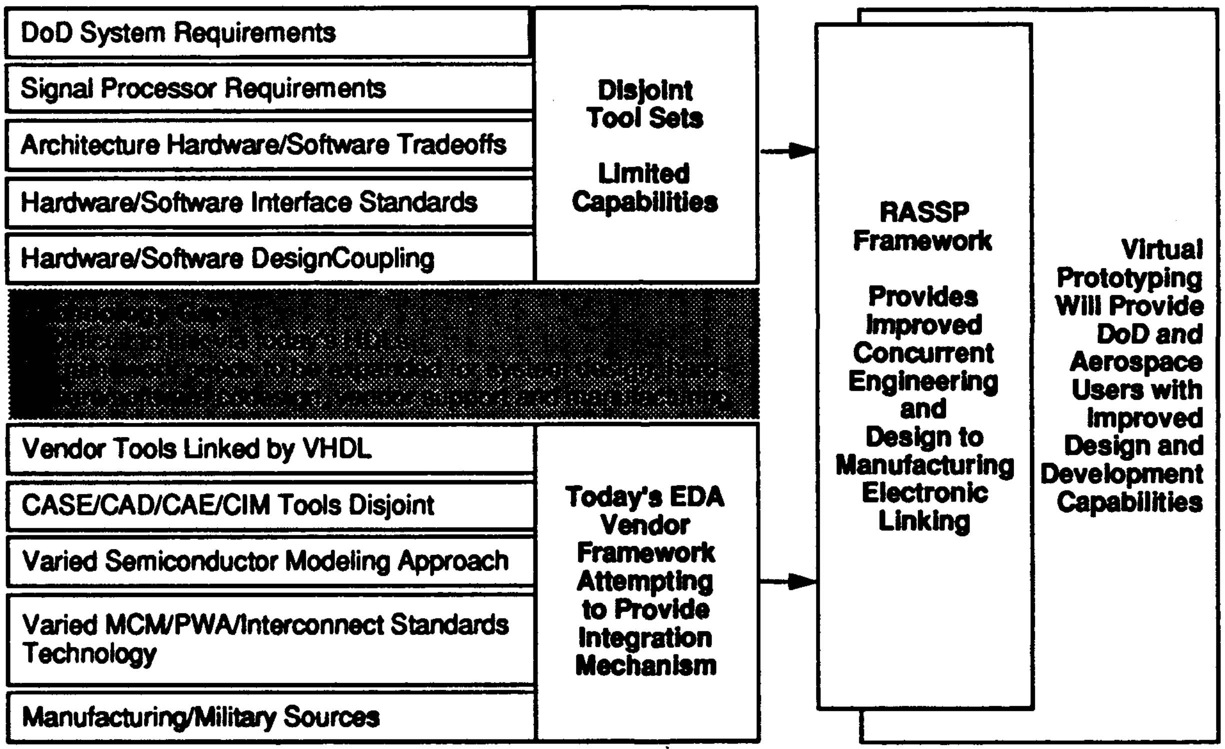


Figure 1-1. RASSP program requirements.

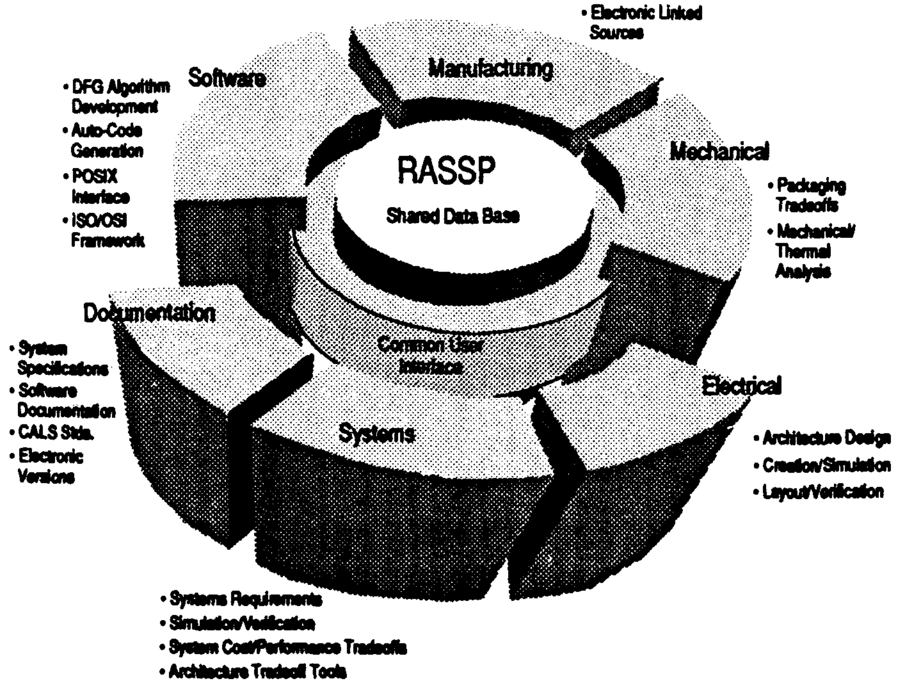
1.3 General Methodology

The GE team has developed a comprehensive and flexible development methodology on the Phase I program. This methodology identifies the top-down design process necessary to support rapid prototyping. This process drives identification of the required CAD tool and system support , the subsequent RASSP development phase will develop the required elements of the system to implement the system, leveraging the commercial developments of the EDA industry.

Key elements of the RASSP system, illustrated in Figure 1-2 include design support for six basic methodology steps: System specification, software development, documentation generation, electrical design mechanical design, and manufacturing and test. The RASSP tool sets associated with these processes are integrated with a framework, providing a common user interface and a shared hierarchical database. This approach will support the design and manufacturing enterprise and will be linked to a broad set of vendors that support quick turn manufacturing.

Details of the RASSP design methodology and the design system requirements are described in Section 5 of this report.

The RASSP system provides a seamless, top-down hierarchical prototyping environment for domain-specific signal processors



Development/Technology Enablers that support:

- **User community**
- **Applications**
- **Infrastructure**
- **Design methodology**
- **Frameworks**
- **Manufacturing**

Figure 1-2. RASSP design system.

1.4 Technical Results

The following mileposts have been accomplished by the GE team on the RASSP study phase:

1. Developed comprehensive design methodology to support RASSP requirements.
2. Identified the key technologies required for further development on RASSP.
3. Linked these requirements to the leading commercial companies and technologists expertise in the required development areas.
4. Developed a Phase II program plan for implementation and demonstration of the RASSP system.

In identification of the rapid prototyping technologies, the GE team examined development methodologies in use at various organizations within GE and external Aerospace companies. This leveraged an extensive body of ongoing work such as GE Engineering Process Improvement (EPI) program, to develop a RASSP recommended development methodology. The GE team then detailed the CAD tool requirements for support of each phase of the methodology, and mapped these requirements with the technology offerings and developments of the industry leaders in EDA tools, research organizations, and associated consortia. A summary of the CAD tool requirements, organized according to functional area is identified in Figure 1-3. The GE focus in development of the RASSP system is to leverage the results of ongoing research and developments to the maximum extent feasible.

System/Architecture Level Design Tools	Detailed Design Tools		SW Generation
System Analysis/Design Top-Level Architecture Design VHDL Support/Extensions Design-For-Test Tools Synthesis Analog Design Tools Reliability Analysis Maintainability Analysis Concurrent Engineering	Digital Hardware Design/Simulation Hardware Modeling Test Generation Fault Simulation Synthesis ASIC Support Tools Gate Array Support Tools MCM/Hybrid Support Tools	Design Rule Verification Tools Signal Crosstalk Tools PWB Tools Chassis Design Tools PWB Support Tools Analog Design Tools Test/Debug Tools Reliability Analysis Power Supply Design	DSP S/W Generation SW Design/ Documentation

Manufacturing	Framework	Support Functions
Manufacturing Advisors Flexible Computer-Integrated Manufacturing Synthetic Manufacturing	CAD Tool Framework Database Management System Libraries/Modeling Simulation Backplane Documentation Support Configuration Management Cost/Performance Estimating Synthetic Environment/Virtual Prototyping	CAD Conferencing Consulting Services Training Services Customer Support Vendor Support

Figure 1-3. RASSP functional requirements assessment functional areas.

GE identified the leading companies/technologists in each of the required development areas, and worked with each company to define the detailed requirements of the RASSP system. Commitments were obtained from each participating organization to support the Phase II program execution, and to work to ensure the widespread acceptance and utilization of the RASSP system.

The GE team developed the initial task/schedules associated with the recommended Phase II program. These were provided to team members associated with various phases of the design process, several technical interchanges were held including a multiple day design review involving all team members, and initial proposals for RASSP developments were received from the subcontractors.

1.5 Important Findings and Conclusions

Many elements of the identified RASSP CAD system are available through commercial EDA vendors, or are already well along in development, primarily by these vendors. To date priorities have been driven by the requirement of the commercial marketplace, which are in most cases consistent with the requirements of the RASSP system.

The keys to the long term success of the RASSP system lie with the endorsement and support of the system by both EDA vendors and the application users (who are, in fact, the EDA customers).

Specific requirements for the long term success of the RASSP system are summarized in Figure 1-4. User community acceptance is tied to the ability of the design system to be integrated with the concurrent engineering methodologies adopted by each particular organization. The RASSP applications need to realize significant benefit in design time savings and upgrade time and cost savings. Demonstrations of the effectiveness of RASSP system for high payoff applications need to be performed. A RASSP infrastructure needs to be established, addressing enterprise wide design and implementation issues, and ties with potential technology suppliers and manufacturing centers. The RASSP design methodology, which determines the requirements for the design system supports both top down design approaches and support of a Model Year. Extensive emphasis is placed on utilization of open systems architectures and interface standards, which are enabling concepts for the Model Year. Advanced hierarchical simulation capabilities are also involved for enabling validation of large complex systems, prior to implementation. Integration of heterogeneous simulations in a distributed backplane is considered one of the enabling technologies for implementation of these large hardware/software systems in a "virtual prototyping" environment.

A common framework is the integration mechanism for the required set of CAD tools and information databases associated with the RASSP design system. RASSP needs to extend the concept of the framework beyond the areas currently being addressed by the EDA vendor community and the CAD Framework Initiative (CFI) to address the enterprise wide issues associated with signal processor design, implementation, manufacturing and integration and test.

Significant development is already underway in automated manufacturing processes, as evident with the systems described in Section 7. Extension of simulation capability to encompass issues associated with manufacturing is required, extending the concept of virtual prototyping mentioned previously. GE Aircraft Engines has developed manufacturing simulation capabilities that are adaptable to meet RASSP manufacturing requirements.

RASSP Hierarchy	RASSP Requirements	Focus Developments
User Community	• Acceptance of User Community	• Concurrent Engineering Coupled to Design System • Enterprise Prototyping
Applications	• Provide payoff to high value applications – Improved capabilities – Reduced design time/cost – Upgradability – Lower life cycle cost	• Proof of Concept Demonstration • Tradeoff Tools—cost/performance, R&M, etc. • Design Reuse Data
Infrastructure	• Provide infrastructure to support wide usage of RASSP system – Data transfer/networking – Centralized libraries/data sharing – Automated manufacturing links	• Commercial Vendor Alliance • Information Management System • Enterprise Electronic Links • Simulation Backplane that Supports Virtual Prototyping
RASSP Design Methodology	• Implement top-down methodology and MODEL YEAR concept – Open system architectures – Commercial technology – Virtual prototyping	• Open Systems/Standard Interfaces • Hierarchical Design for Test • Processor Virtual Prototyping • Model Year Concept
RASSP Framework	• Develop hierarchical, seamless design framework • Develop process, as well as design models	• Top-Down Design Tools – System Design Tools – Design Advisors/Synthesis – Design Language Extensions (HDL & SDL) – Hardware/Software Codesign • Hierarchical Framework – Common User/Tool Interfaces – Tool Hierarchy Definition – Common Data Representation
Manufacturing	• Automated Manufacturing & Test	• Process Modeling/Simulation (Virtual Prototyping)

1.6 Significant Hardware Development

The initial analysis of the RASSP requirements indicates that significant improvement in the design and manufacturing cycle can be made without significant hardware development as part of the RASSP implementation phase. Hardware design items for consideration for RASSP are in the areas of simulation accelerators relative to the design system, and for support of the application demonstrations. It is anticipated that the computational and information storage needs of the RASSP system will be adequately addressed by the commercial computer suppliers.

Hardware development likely on the program is for support of a application demonstration, the details of which need to be identified in conjunction with the government. Potential applications to be considered for demonstration are described in Section 8. The RASSP study considered electro-optical interconnect concepts and decided that the ongoing work being sponsored by DARPA and DoD adequately addressed the requirement.

The development of equipment that would support accelerated prototyping of semiconductor parts, MCMs and printed wiring assemblies is being addressed by other ongoing programs at DARPA and Tri-Services (ex: ASEM, MMST, etc.); therefore, GE has not emphasized fabrication equipment. However, in the test area there are areas where extensions to support analog and digital testing by the same equipment may be an area worthy of investment.

1.7 Special Comments

In order to achieve the maximum benefit from the RASSP system for the Aerospace industry, the GE team believes that the program is best managed by a large aerospace contractor with a sensor-based systems perspective, and is best executed by a hand-picked team of industrial organizations, EDA tool developers, research organizations/universities, and consortia. It is also recommended that several user organizations participate in the program to ensure accommodation of various design methodologies, and various application requirements. The GE proposed approach for Phase II is to make the design system available to other Aerospace companies as alpha and beta sites.

1.8 Implications for Further Research

Further research is required in selected high priority technology areas, in addition to the recommended development areas, as shown in Figure 1-4, to enable the full potential in cost and schedule savings on signal processor development programs using RASSP.

Several of these high priority research areas have been identified by the RASSP study program. Some examples are shown below:

Simulation Backplane Technology - Enabling validation of large hierarchical simulation models, with various information representations, timing schemes, and control structures.

Design-for-Test approach that applies to all levels of design.

Design Advisor Technology, with particular emphasis on system level design. Technology for hierarchical management of design advisors is also required

Further details of these research areas are included in Section 6.0, as well as the recommended RASSP development areas.

1.9 Standard Form 298, February 1989

Standard Form 298 follows on the next page.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE October 21, 1992

3. REPORT TYPE AND DATES COVERED Technical Report 5/18/92 - 10/21/92

4. TITLE AND SUBTITLE
Rapid Prototyping of Application Specific Signal Processors (RASSP) Program - Study Phase Final Technical Report

5. FUNDING NUMBERS
CMDA972-92-R-0017

6. AUTHOR(S)
John Welsh, James Saultz, Jeffrey Pridmore, et. al.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
GE Advanced Technology Laboratories
Moorestown Corporate Center, Route 38
Moorestown, NJ 08075

8. PERFORMING ORGANIZATION REPORT NUMBER
0002AB

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
DARPA/ESTO
3701 North Fairfax Drive
Arlington, VA 22203-1714

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
None.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
No limitations.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The objective of the RASSP program is to develop the required methodologies and supporting CAD systems, to enable the defense industry to rapidly specify, produce, and yield domain-specific, affordable signal processors for systems such as automatic target acquisition, tracking and recognition, electronic countermeasures, communications, and SIGINT".

The GE accomplishments on the program include: Identification of the key rapid prototyping technologies required for further development , identification of the leading commercial EDA companies and technologists with expertise in the required development areas, and development of a program plan recommendation, for implementation and demonstration of the RASSP system.

The GE team examined development methodologies in use at organizations within GE Aerospace, and several outside companies, and leveraged extensive work performed in the GE engineering process improvement program, to develop a RASSP recommended development methodology. The GE team then detailed the CAD tool requirements for support of the methodology, and mapped these requirements with the technology offerings and developments of the industry leaders in EDA tools, research organizations, and associated consortia. The required development areas, which are not anticipated to be addressed by the EDA CAD industry, with addition of application demonstrations, form the basis of the GE recommended program plan for RASSP.

14. SUBJECT TERMS

15. NUMBER OF PAGES 150

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unlimited

2. UNDERSTANDING THE PROBLEM

2.1 Background

The challenge of designing application-specific signal processors requires a multi-discipline concurrent engineering approach. Signal processor design and implementation are typically driven by two major items - system programmatics and system requirements.

- Programmatic issues drive major decisions based on cost, schedule, and risk constraints. Program managers do not want to put their programs at risk due to unproven technologies and designs.
- System requirements for sensor systems are often driven by implementation constraints (size, weight, and power) versus performance (processor throughput).

Resolving these two issues requires informed decisions by system designers in the initial program phases. These decisions, typically made during the first ten percent of the design cycle, often determine ninety percent of the total system costs. Support for early decision making is an area that is currently deficient and demands concerted effort.

Industry's typical approach for developing signal processors is to build upon existing custom designs by adding hardware and software modules. The advantage of this approach is that existing designs have a heritage of qualified parts, software modules, and programmable support tools for generating new code. The disadvantage is that the fielded systems are no longer state of the art, and do not have adequate mechanisms to enable low cost technology upgrade.

The RASSP methodology needs to address this dilemma by providing a comprehensive design system and methodologies to support rapid design and fabrication of application-specific signal processors on a MODEL YEAR basis, thus enabling regular technology updates with minimum hardware and software breakage. This goal can be supported through the use of COTS processor technology and its associated support software.

2.2 GE Engineering Process Improvement Program

In early 1989, GE Aerospace began an ambitious program aimed at developing state of the art processes in key engineering disciplines. The Engineering Process Improvement (EPI) program has been implemented in all 13 GE businesses and has begun to provide significant productivity gains, which are summarized in this section.

The program was initiated from a recognized need for improvement in the engineering processes in order to significantly reduce product development costs and cycle time, goals which are consistent with the RASSP program.

Engineering costs represent a relative high percentage of product costs for many military systems as shown for the GE Aerospace group in Figure 2-1, hence substantial improvement in product costs are realizable through the use of CAD tools.

More importantly, decisions made by engineering early in the product development phase, have a very significant effect on the overall product costs. As indicated in the graph on the right of Figure 2-1 shows that at the completion of concept design phase, 5% of the costs have been incurred, while 60 percent of the product costs have already been committed. At the completion of prototype testing when only 15% of the total costs are incurred, nearly 90% of the program costs have been committed, leaving in this case just 10% of the costs that can be affected by subsequent decisions.

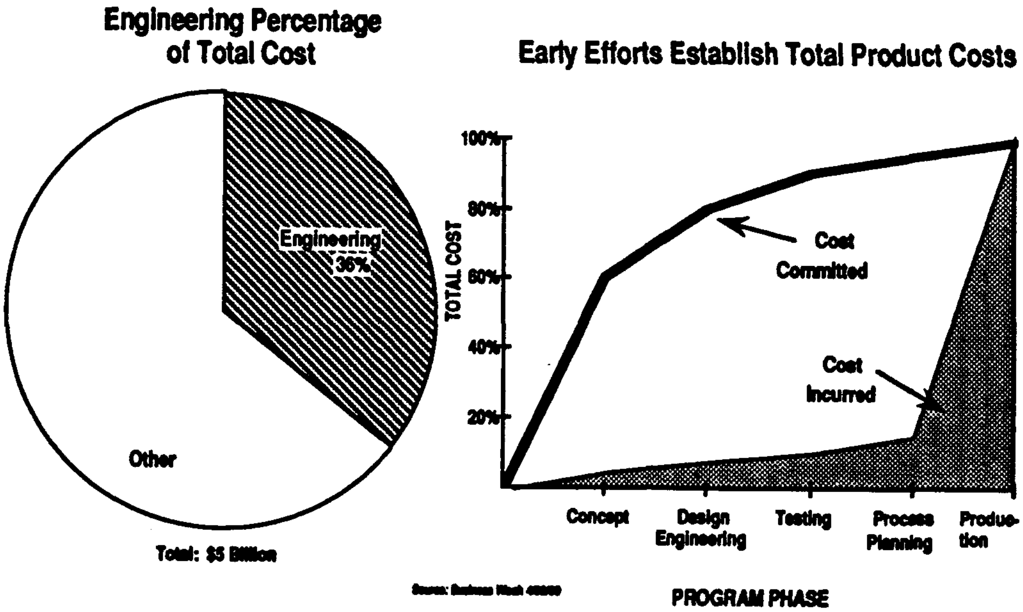


Figure 2-1. Engineering cost summary relative to program phase.

On the EPI program, after approximately a year of study by hundreds of individuals, representing all of the GE engineering groups, approximately 100 best practices were identified in 10 best practice areas (Figure 2-2), and documented. These practices, form the basis for development of the EPI design methodologies. The best practices were detailed, and approaches for utilization of automation in the processes through use of MCAD, ECAD, and CASE were investigated by EPI engineering subcouncils. These investigations produced tool requirements for supporting each of the disciplines.

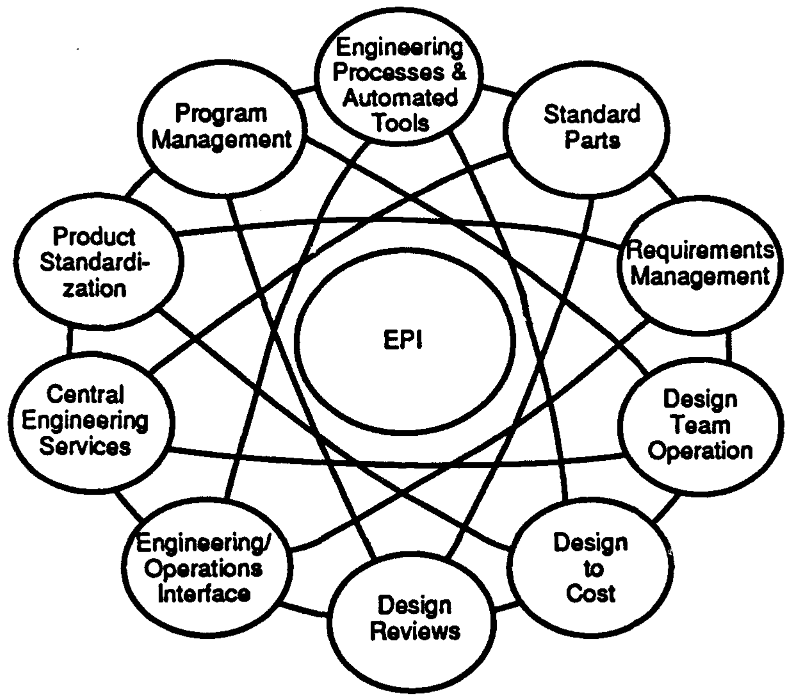


Figure 2-2. EPI best practices.

Thirteen suppliers have been selected to cover the spectrum of tool requirements, that are key to the EPI strategy, the tools were standardized across the 13 businesses. The results were greater proliferation of these tools than otherwise would have been possible due to significantly lower procurement costs, common training, greatly reduced support tasks, and group maintenance agreements. The GE standard tool selections and the processes to which they apply are illustrated in Figure 2-3.

Figure 2-4 indicated the number of workstations and software development seats purchased, since the first EPI purchase in March 1990. These do not represent the total number of workstations and tools at GEA, as the installed base prior to EPI is not included.

Preliminary results gathered on GEA programs using EPI methodologies shown in Figure 2-5 indicate substantial savings achieved in all design disciplines, even at this early phase in the implementation of the EPI program.

The GE team has initiated discussions with several large aerospace companies that will be users of the deployed RASSP design system (both RASSP Phase I participants and non-participants). The one common finding in discussions with the large aerospace companies was that they all desired a RASSP-like design system. In fact, they would have liked to see the commercial suppliers of design systems offer such a system with a part number. This common feedback was based on the results of most companies attempting to provide a RASSP-like system based on buying the various pieces and then trying to integrate them. Some companies indicated that the cost of integration was three to five times the cost of the CAD software.

RASSP will build upon the GE-EPI installed tool experience. Standard design tools have been selected to support each engineering discipline.

	Systems	Software	Digital	Analog	RF/Microwave	Mechanical
SUN Microsystems	●	●	●	●	●	●
CADRE Technologies	●	●				
Expertware		●				
GEC Marconi	●					
Ascent Logic	●					
Comdisco Systems	●					
Mentor Graphics			●	●	●	
Teradyne			●			
Analogy				●		
EESof					●	
SDRC						●
Interleaf	●	●	●	●	●	●
Ingres	●	●	●	●	●	●

Figure 2-3. GE-EPI tool selections.

	Seats Planned #	Program to Date 6/30/92		1992		
		Seats Procured #	Expended to Date $M	Plan $M	Actual $M	% of Plan
Workstations	3500	3181	46.4 *	15.2 **	14.1 **	93
Systems	528	184	1.8	1.0	1.1	110
Systems/Software	920	660	4.7	0.5	0.5	100
Software	468	449	1.5	0.2	0.2	100
Digital	580	683	9.7	2.7	1.4	52
Analog	900	58	1.2	0.5	0.2	40
Microwave	50	73	1.6	0.2	0.5	250
Mechanical	250	205	5.3	1.5	1.3	87
Support Software	N/A	N/A	1.0	0.6	0.3	50
			73.2 *	23.4 **	19.6 **	84

* Included $11.2M purchased under contracts
** Included $6.9M purchased under contracts

- **Dollars include purchase of products, maintenance, and training.**
- **Purchases have been negotiated at substantial savings.**
- **Most tools were mature proven tools.**

Figure 2-4. GE-EPI design system hardware and software cost to date.

Discipline	Business	Program	Engineering Design Activity	PreEPI	Reduction
Analog	ACS	C17	PWB - Electrical	1144 hrs	44%
			Product Design	328 hrs	83%
			Drafting	725 hrs	90%
	GES	BSY-2	Surface Mount/Hybrid Des	144 hrs	50%
	GES	AN/SPY-1B	PWB Design	311 hrs	48%
	ASTRO	EOS	Battery Power Conditioner	3.3 mos	30%
	GCS	IRR	Circuit Simulation	3 hrs	99%
Digital	DS	FBM	ASIC Design	4300 hrs	44%
	AES	AADEOS	PWB Design Cycle	12 wks	50%
	AES	IRST, GD53	Test Vector Generation	8 wks	25%
	GES	AN/SPY-1B	PWB	324 hrs	56%
Mechanical	AS	RAH 66	Rapid Prototyping of Mechanical Parts	200 hrs	85%
	ASTRO	P91B (Prop)	Spacecraft Structural	48 wks	38%
	ASTRO	EOS	FEM Mass Prop	510 hrs	53%
	ASTRO	INTELSAT	C-Band Feed Section Design, Fab, Test	5 wks	40%
	SCS	Visual Display System	Mechanical Structure	1448 hrs	47%
	GES	COBRA	Antenna Structural Analysis	144 hrs	75%
	RES	Endo LEAP/SCSM	Rapid Proto Mechanical Parts	200 hrs	85%
Microwave	ASTRO	IR&D	Amplifier Design B/B	1 mo	Eliminated
	ASTRO	Telstar 4	C-Band Beacon Transmitter	2 mm	25%
	GES	COBRA	Design of $RF Transitions for T/R Module I/O	3 mos	90%
	AES	Proj 621 (A12)	Semi-rigid Cable Design	30 hrs	50%
	ASTRO	ATDRSS	Comm Subsystem Design	10 wks	20%
Software	AES	AADEOS	Ada Software Dev.	57 wks/ 1000 LOC	30%
	O&RS	AN/BSY-2	Acoustic Processing	59 wks/ 1000 Loc	28%
	SCS	VISIONIC Data Base	Generate Database for Visual Simulation of Terrain	270 hrs (3600 sq. nautical miles basis)	44%
Systems	ASTRO	INMARSAT	Doc. of Subsystem	10 person-months	25%
	GCS	Nerve Trunk	Generate System Specification	3 mos	50%

Figure 2-5. GE-EPI experience on cost and schedule savings.

The other area where most of the aerospace companies agreed on was that the only areas where CAD tools were somewhat seamlessly integrated was in the lower level ASIC, MCM and PWA areas.

None of the large EDA vendors (Mentor, Cadence, DAZIX, etc.) provide CASE tools as an integrated set. In fact, most of the companies started with CASE tools and have abandoned them. Current discussions with EDA vendors leads GE to believe that they are moving back towards integrating CASE tools into their offerings.

During the RASSP Study Phase, NAVAIR had a procurement for buying a set of electrical and mechanical design tools. GE discussions with the EDA vendors indicates that vendors started to understand the problems industry has when they purchase tools and integrate them. This experience should help the EDA companies that have been primarily organized on a product basis to group the need for an integrated approach that spans across all products. GE has worked closely with Mentor on defining the approaches to integrating a broad set of tools and believes the experience will allow the RASSP implementation phase to move much quicker in the early implementation phase.

<u>Lesson Learned on EPI Program Implementation</u>

The scope of the program was broader than envisioned in 1989 when it was started. It became clear later on that if we were going to improve productivity in engineering, we were going to have to break down organization barriers and involve other organizations in the program. Unifying our processes across business functions is key to achieving substantial improvements in productivity such as cycle time reductions.

An infrastructure was essential for a change of this magnitude. Top down drive of the Managers of Engineering was necessary; but the use of subcouncils helped an empowered work force to accept the changes and now facilitate and improve the processes.

A good set of Design and Manufacturing Standards provides a basis for implementation of concurrent engineering practices and producibility engineering. Developing the standards jointly with manufacturing gives buy-in by both organizations.

Parts standardization was more involved than anticipated. Implementation is easier on newer programs than on existing programs where the customer already has established a logistic support capability. Most customers even on new programs have their own program preferred parts lists which, as you would expect, are different.

The development of a library management system (LMS) with a large set of COTS and custom parts that were supported by models that could be used in the simulation was a significantly bigger job than was anticipated.

Measuring progress is essential for continuous process improvement and has helped to keep the program sold. We have been able to show that the payback is exceeding the investment.

Documentation on EPI processes consists of methodology documents and tool documents. This information is maintained by the GE Engineering Support Center. A listing of available documentation is provided in Table 2-1.

The RASSP study phase has taken full advantage of the EPI lesson learned and will be able to adopt or use much of the approach developed under the EPI project.

Table 2-1. EPI documentation.

General Engineering/Manufacturing

DOC. #	DOCUMENT TITLE	LATEST REV.
100-01	Release Standards	
	Section 1- PWB/CCA	2/28/91
	Section 2- Castings, Machinings & Sheet Metal	6/9/92
	Section 3- Cables and Harnesses	3/6/92
	Section 4- ASIC	2/28/91
	Section 5- Engineering Parts List	2/28/91
	Section 6- Part Requirements	6/9/92
	Section 7- Serial Numbers	6/9/92
	Section 8- Change Notices	6/9/92
100-02	PWB/CCA Design and Manufacturing Standards	2/28/92
100-03	Engineering Metrics	9/15/91
100-04	Cable and Harness Design & Manufacturing Standard	2/28/92
100-05	Configuration Management Process	1st Rel., 11/91
100-06	GEA Schematic Guidelines	Rev.1.0, 6/17/91
100-07	Design-to-Cost Methodology Handbook	Rev.1.0, 7/15/91
100-08	Concurrent Engineering Manual	Draft-3/1/92
100-09	Casting & Machining Design & Manufacturing Standard	Draft-10/92
100-10	Backplane Design & Manufacturing Standard	Draft-10/92
100-11	Ceramic Module, Multichip Module, & Hybrid IC Design & Manufacturing Standard	Draft-10/92

Engineering Process/Methodology - Hardware

DOC. #	DOCUMENT TITLE	LATEST REV.
200-01	Digital Engineering Process	1/15/91
200-02	Instructor Guide for Digital Engineering Process	1/15/91
200-03	Student Workbook for Digital Engineering Process	1/15,/91
210-01	Analog Engineering Process	Ver. 2.1, 12/17/90
210-02	Instructor Guide for Analog Engineering Process	Original
210-03	Student Workbook for Analog Engineering Process Training Course (Vols. I & II)	Original
220-01	Mechanical Engineering Process	Original
220-02	Instructor Guide for Mechanical Engineering Process Training Course (Vols. I & II)	Original
220-03	Student Workbook for Mechanical Engineering Process Training Course (Vols.I&II)	Original
230-01	RF/Microwave Engineering Process	Original
230-02	Instructor Guide for RF/Microwave Engineering Process Training Course	Original
230-03	Student Workbook for RF/Microwave Engineering Process Training Course	Original
240-01	Day One Instructor Guide	4/91-Rev A
240-02	Day One Student Workbook	Original-2/91
250-XX	Not Assigned	

Systems/Software

DOC. #	DOCUMENT TITLE	LATEST REV.
260-01	Software Engineering Methodology Handbook	Ver.4.0, 1/17/92
260-02	Instructor Guide for Software Engineering Methodology Training Course (Vols. I & II)	Rev. B, 7/15/91
260-03	Student Workbook for Software Engineering Methodology Training Course (Vols. I & II)	Rev. B, 7/15/91
270-01	Systems Engineering Methodology Handbook	Rev. 1, 1/3/92
270-02	Instructor Guide for Systems Engineering Methodology Training Course (Vols. I & II)	Original

270-03	Student Workbook for Systems Engineering Methodology Training Course (Vols. I & II)	Original
270-04	Systems Engineering Training Course Reference Book	10/1/90
280-01	Systems/Software Engineering Process Model	Ver. 3.0, 2/8/91
290-01	Day One (Software) Instructor Guide	Original, 2/91
290-02	Day one (Software) Student Workbook	Original, 2/91

CAD/CAE Tools & Design Support

DOC. #	DOCUMENT TITLE	LATEST REV.
300-01	Geometry Standard	Rel. No. 1, 9/3/91
300-02	GEA Library Standard for Mentor Graphics LMS	Rel. No. 1, 6/1491
300-03	Aerospace Preferred Parts List (APPL)	Rev. D, 6/92
300-04	General Electric Digital Process Tools Course Student Workbook	Original
300-05	Standard Parts System Users Guide	Ver. 1.1, 6/92
300-06	Standard Parts System Student Workbook	Ver. 1.1, 6/92
300-07	Standard Parts System Instructor Guide	Ver. 1.1, 6/92
300-08	Tool Training Guide	2nd Edition, 11/91
300-09	Tool Training Price Guide	1st Edition, 11/91
300-10	Library Management System Users Manual for Design Engineers	1/92
300-11	Library Management System Requirements Document for MGC V8 Software	Rev. 1.1, 12/5/91
300-12	SABER Interface to GEA LMS	Rev. 1.1, 4/92
300-13	Digital Integration Demonstration Vehicle	Rev. 1, 12/16/91
300-14	Simulation Hardware Accelerator-Requirements Document	Rev. 2.0, 10/31/91
300-15	Simulation Hardware Accelerator-Benchmark Testing Document	Rev. 1.1, 2/17/92
300-16	GEA SUN Configurations Guidelines	Rev. 1.0, 2/92
300-17	GEA Archiving Tool System Requirements Document	Ver. 1.1, 9/26/91
300-18	Worst Case Timing Simulator Requirements Document	Ver. 2.2, 2/20/92
300-19	Mechanical Integration Demonstration Vehicle	Original, 5/16/91
300-20	Mentor V8.0 Acceptance Test Specification	Draft, 7/9/92
300-21	GEA CAE/CAD Requirement Document	Ver. 2.0, 2/20/90
300-22	GEA RF/Microwave CAE/CAD Technical Requirements	Ver. 1.3, 5/20/91
300-23	Microwave Integration Demonstration Vehicle	Ver. 1.0, 12/31/91
300-24	Interleaf Integration Demonstration Vehicle	Draft, 1/23/92
300-25	GEA PWB Geometry Library	Draft, 3/92
300-26	SPS - LMS Interface User Documentation	Original, 6/92
300-27	Designing With Mentor Graphics V7 Software Using LMS	Ver. 1.0, 7/8/92
300-28	LMS Reference Guide	On hold
300-29	SPS Database Administrators Guide	Draft, 9/92
300-30	GEA LMS V7 to V8 Library & Design Conversion	Draft, 9/92

3. GE TEAM

GE believes the key to RASSP success is assembly of a world class team, composed of leaders in all the required disciplines to execute the program. The GE strategy on the Phase I program was to develop and maintain a large team, covering all aspects of the RASSP design requirements, leveraging investments at many organizations, and cultivating competition in key development areas. The complement of organization type and associated areas of expertise required to address the RASSP program is indicated in Figure 3-1. The model shown in Figure 3-1 has been used in the study phase and will continue to be used in Phase II.

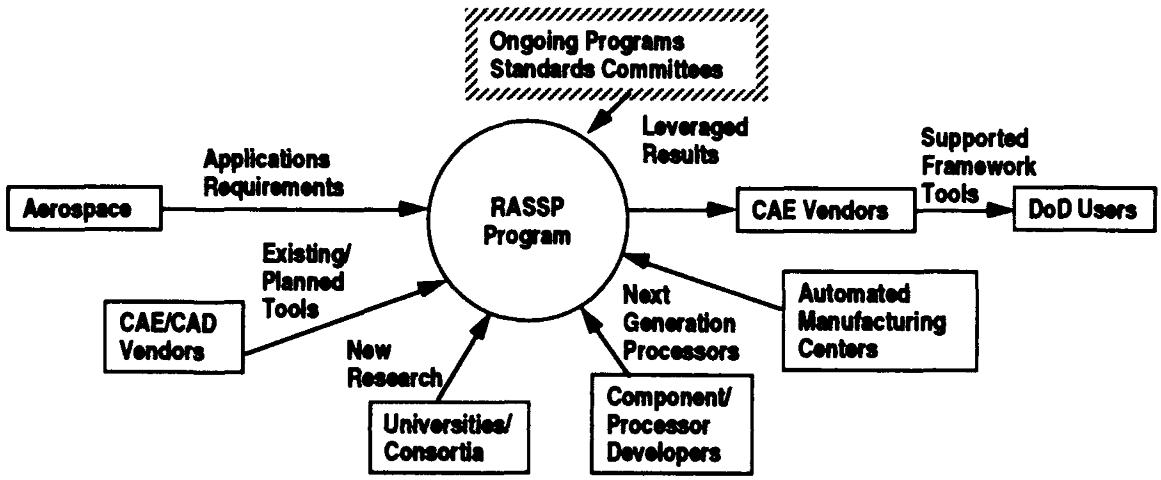


Figure 3-1. GE teaming strategy.

For the Phase II program, a more focused team will be selected, based on development capabilities, existing and in progress technologies, and cost.

The GE team, which has continued to evolve over the course of the study program, consists of organizations in the following general categories, as indicated in Figure 3-2. Each team member has unique skills in their respective disciplines, and offers excellent potential for the RASSP Phase II program. Brief summaries of their capabilities and anticipated contributions to the RASSP program are provided below. More detailed information on the organizations and proposed concepts for the RASSP implementation phase is provided in Section 10 of the report.

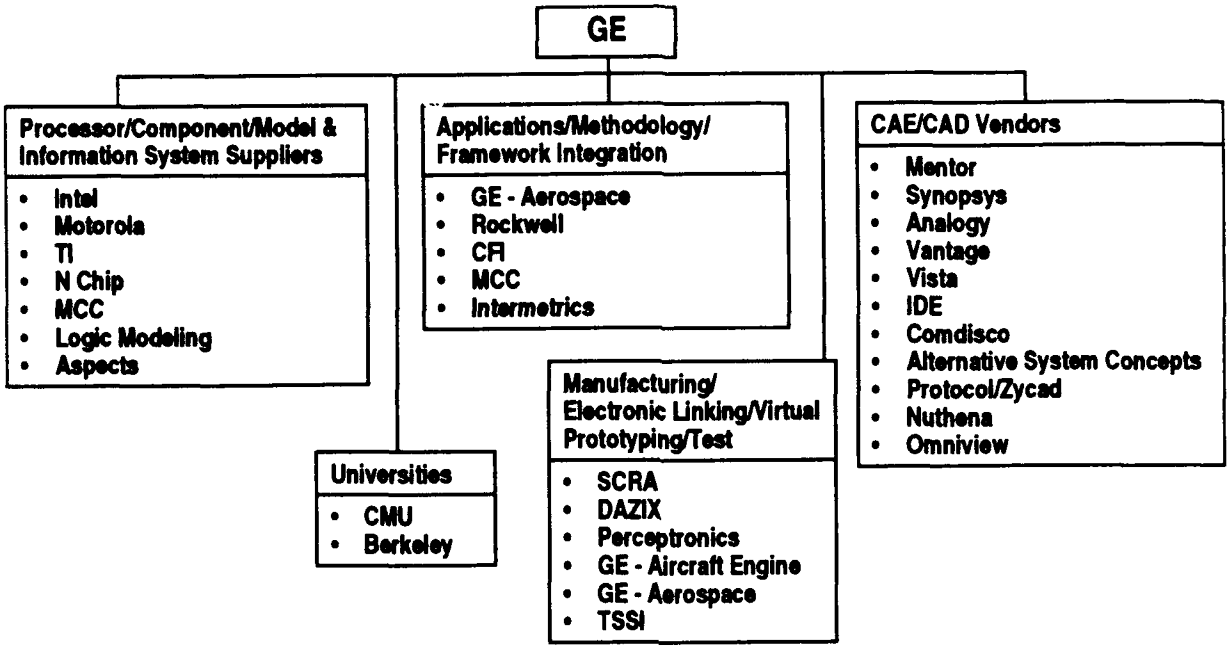


Figure 3-2. RASSP program team.

In the area of microprocessor technology, Intel and Motorola represent the market leaders, while TI is the leader in the single chip digital signal processor market. In the MCM market, TI, Motorola, IBM and nCHIP are leaders in the MCM market. All of the MCM vendors mentioned have supported the study phase except IBM.

In the area of modeling technology, Logic Modeling is a leader in component modeling and hardware modeling, Aspects and Mentor hold strong positions in component information management systems (library management).

GE Aerospace, and Rockwell which are leading DoD and NASA contractors in the design and manufacture of electronic systems (from large highly sophisticated systems to low cost high volume systems), offer significant capability in requirements development, methodology definitions and management, system integration, and application demonstrations. Significant internal investment in GE's ongoing Engineering Process Improvement program (EPI), described in Section 2.0, has already developed near term solutions for many of the core technology and management areas relative to RASSP. These developments will be made available to the RASSP program, enabling the RASSP resources to be applied to critical development areas.

The CAD Framework Initiative Organization (CFI), a public organization, was formed in 1988 out of a recognized need in the CAD community for establishment of standards to enable interoperability of tools. This organization offers the team significant expertise in establishing the requirements for extended capability frameworks to address the RASSP requirements.

Intermetrics has been the recognized leader in the development of hardware descriptive languages (VHDL and MHDL), and hence is well qualified to address the extended information representation requirements for RASSP. Intermetrics will be supported by Analogy on analog HDL developments.

Carnegie Mellon University, and University of California at Berkeley, are highly regarded research organizations, with significant ongoing programs closely related to the RASSP objectives. CMU has particular strengths in synthesis technologies, and high level design tradeoff advisor tools. Omniview is currently making CMU synthesis technology commercially available with links to commercial EDA tools. Berkeley is a leader in advanced codesign concepts, tools and a framework that supports codesign.

South Carolina Research Authority is a recognized leader in the development of flexible computer integrated manufacturing technology, and is the prime contractor for the Navy's Rapid Acquisition of Manufactured Parts (RAMP) program. SCRA, in conjunction with the GE automated manufacturing centers offer excellent capability to address the automated manufacturing, test, and electronic integration requirements of the RASSP program. GE has worked with Mentor to address integrating CIM tools from Mitron into the RASSP system. Mitron provides software that supports pick and place and other manufacturing equipment.

DAZIX/Intergraph is a leader in electronic information management systems, and a prime supplier of design tools to NAVSEA, and is heavily involved in CALS programs. In addition, DAZIX is a leading supplier of electronic design CAD tools. DAZIX has supplied a technical information management system to NASA for support of the Space Station program.

GE-Aircraft Engines, is a leader in development and application of concurrent engineering concepts, factory simulations, and rapid prototyping concepts. GE-Aircraft Engines is also the prime contractor on the DARPA Initiative in Concurrent Engineering program, and will make results/developments available to RASSP for the mutual benefit of both programs.

TSSI offers test development tools, supporting utilization of EDA vendor test information at all levels of design and test in the hierarchical development process. The TSSI tools when coupled with the RASSP design and manufacturing system will provide a virtual test capability.

Several EDA CAD vendors are supporting the GE RASSP team, each with particular and potentially overlapping areas of expertise. Mentor Graphics is the number one vendor in the overall electronic design CAD industry, with product offerings covering most design areas, and a supporting framework, and alliances with other vendors for tool integrations and cooperative developments. Synopsys is the leader in offering synthesis technology, enabling correct by construction designs for a variety of ASIC technologies, which are comparable or better in critical performance/size parameters relative to designer generated implementations. Synopsys recent selection by DARPA on the ASEM program to develop synthesis tools for MCM designs will support the RASSP development system.

Analogy is a leader in analog design tools and analog hardware descriptive language developments with their MAST product. They have also integrated their products with the Mentor framework and tool sets.

Vista provides a strong basis in VHDL tool sets and VHDL language developments.

IDE's primary focus is in software development support tools for design and analysis. CASE offerings are also of particular interest to RASSP.

COMDISCO is a leading supplier of system level design tools for supporting the top level design phases for signal processors. These include network simulation tools, signal processing algorithm and implementation support toolsets.

NuThena is also a leading supplier of system level design tools, with unique capabilities in high level design capture and modeling tools. New thrusts are also on-going in synthetic environments and distributed interactive simulation.

Alternative System Concepts is a new company focused on design for test tools based on utilization of VHDL.

Protocol/Zycad is also a leader in development of design for test concepts, synthesis technology for test implementations, and accelerator technology.

MCC has recognized capabilities in development of design advisors, design for test concepts and programs, MCM technology and known good die approaches. MCC's recent selection by DARPA to conduct the ASEM-MCM alliance role will provide valuable inputs to guide the RASSP design and implementation phase.

Vantage has been and continues to be a leader in the development of VHDL simulation. Vantage's support on developing VHDL extensions and support for simulation backplane approaches will contribute to meeting the goals of RASSP.

Mentor's recent selection by DARPA on the ASEM program to conduct a significantly improved MCM placement and routing program will help RASSP MCM designs.

GE has initiated discussions with Teradyne and Quad Design to analyze how their tools integrated with Synopsys tools will support the ASEM program and act as an integrated set of tools that can be coupled into the RASSP design system to provide an improved design and manufacturing capability.

4. MODEL YEAR CONCEPT

Definition: The RASSP system must support processor upgrades each MODEL YEAR, providing substantial improvement to the overall system performance (e.g., 40%) in many cases without requiring re-work of either hardware or software portions of other parts of the overall system. A MODEL YEAR processor may take longer or shorter than a calendar year to produce, but it is anticipated that each MODEL YEAR design has the potential to be shorter in duration than the preceding one as design and fabrication capabilities mature.

Model Year Concept - New System Technology Leverage: Commercial processor technology offers significant capability upgrades every two to three years, yet typical military development cycles are over five years in duration to deployment. The processor technology increasing performance and downward cost progression are evident in Figure 4-1. Performance is indicated by the dashed lines, while the solid lines indicate the cost trends. Regular performance increases of more that 2X are indicated for both scaler and vector processors on a two year basis This results in a situation where the technology in the fielded system is one to two generations behind the state of the art at the time of deployment of the system. Figure 4-2 illustrates the relationship of model year upgrades to the military equipment development cycle. It is evident that the design must accommodate technology insertion consistent with the Model year design concept, in order to achieve the maximum benefit of the available technology at the time of deployment.

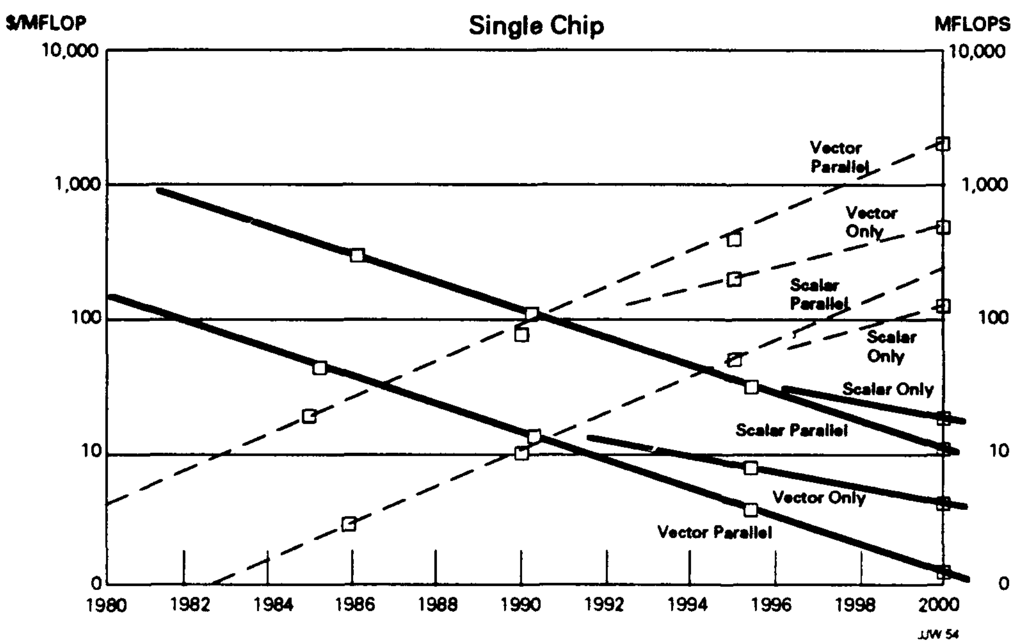


Figure 4-1. Commercial processor performance trends.

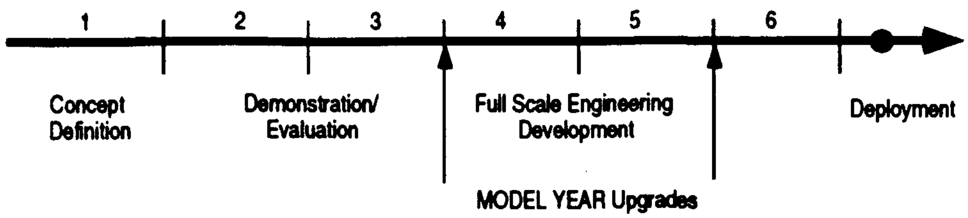


Figure 4-2. Model year concept enables technology leverage at deployment.

Other key factors in the current equipment development environment which drive the need for a model year upgrade approach are as follows:

- Decreasing military budgets are forcing a heavier reliance on commercial processing technology where practical
- Ruggedized equipment versus full militarized hardware is becoming more prevalent in new procurements
- Competition in the commercial processor business has led to acceleration in the pace of new processor releases.

Model Year Concept - Life Cycle Cost Issues: The normal platform life cycle for military systems is often more than twenty five years. In addition the current DoD focus is on further extended life cycles, beyond the original design plans, for budgetary reasons.

These systems hence undergo a potential of 6 - 8 technology upgrades (probably 3 to 4 actually get exercised) over the operational lifetime, as illustrated in Figure 4-3, both to address performance issues as well as technology obsolescence.

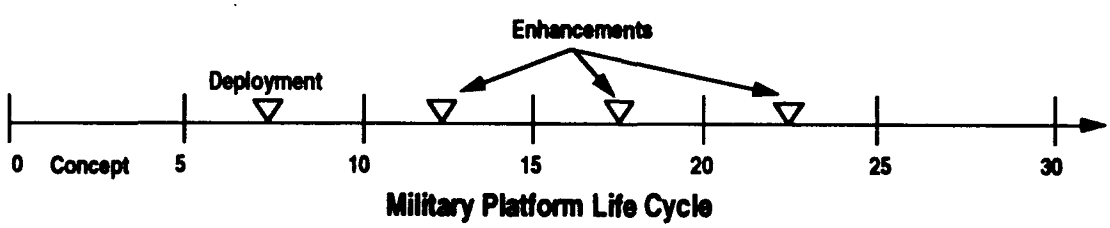


Figure 4-3. Model year enhancements over system life cycle.

The model year upgrade approach will provide substantial savings in multiple aspects of the technology upgrades. Standard interfaces will likely reduce the hardware upgrade cost by a factor of 3, by minimizing the amount of the redesign required for the upgrade, and by reduction in the integration via utilization of standard, proven, and already debugged interfaces. Utilization of designs captured in HDL's, with supported tools will also contribute to reduced hardware upgrade costs. Software expense associated with upgrades can be reduced by an even larger factor (4 to 8) as a result of 1) software reuse (retargeting of application code), 2) automatic regeneration of signal processing code using RASSP system level tools, 3) use of standardized software (ISO/OSI) interfaces, and 4) reliance on commercial operating system (microkernal) technology.

Other specific aspects of the model year concept contributing to life cycle cost reduction are as follows:

- The ability to readily upgrade the system will reduce the number of spare units required, particularly in situations where parts become obsolete, and organizations typically make lifetime purchases prior to the part going out of production.

- Utilization of COTS technology reduces the burden of logistical support in that vendors can be expected to maintain inventories of products.

- Software support costs are amortized over a larger market, as a result of utilization of commercially supported operating systems and interfaces.

The focus of the model year on utilization of standards and COTS technology however does result in an achievable system performance which is less that what would be achievable with a customized design approach by a factor of 2X to 4X (based on available COTS technology today). The degree of this impact is decreasing with the increasing technology performance trends. This performance impact versus NRE cost is a tradeoff that must be addressed during the initial system level tradeoffs within the RASSP system. Example of processor design based on COTS processing that offers significant performance are the Aladdin and Touchstone designs. GE is evaluating both of these designs for application to the RASSP implementation phase.

<u>Model Year Concept - Definition/Basic Principles</u>: The Model Year concept is a key element of the RASSP design methodology which is the enabler to allow systems to realize the benefit of low cost technology insertion for each initial deployment, and over a product's life cycle cost, as mentioned in the previous sections. The Model Year concept is based on application of the open architecture design principles in the development of equipment. Adherence to these principles is ultimately up to the particular design team, however the RASSP methodology and supporting design system provide the necessary tools and guidelines. These detailed definition and implementation procedures will be formalized on the RASSP development program (similar in principle to the design to cost procedures developed for the GE engineering improvement program described in Section 2) and include:

- A set of standard hardware interfaces for use at all the levels of signal processor interconnection. This includes serial interfaces, bus interfaces, point to point parallel interfaces, etc. Interface selections will offer a variety of performance/cost characteristics.

- Standard interfaces will support ISO/OSI to provide clean hardware/software implementation interfaces, and to minimize software breakage on technology upgrades.

Standard Test Methodologies should be selected for utilization at various levels of design. Approaches for automatic, or advisor based generation of test hardware and software are utilized, enabling high efficiency in this aspect of development, hence encouraging completion of the test concept design and implementation early in the development cycle.

- Models of new processor or other technologies will be made early in the product life cycle to enable development/upgrade of model year designs in anticipation the new technology releases.

- Modular design approaches will be utilized to enable incremental upgrades to systems via addition of elements, as well as technology upgrades.

- Support software will be developed in high level languages, to enable maximum practicality for reuse. Object-oriented programming approaches per signal processing algorithms will also be supported.

- Automatic code generation for application code generation and documentation will be utilized for peak efficiency in design changes, and technology upgrades. CASE tools for automatic documentation, will be used to address applicable military documentation requirements. Code generators will generate portable high level language, enabling portability to multiple processor types for validation.

Model Year Architecture Concept: The RASSP hardware architecture, implemented with open systems concepts mentioned previously, will have the generic format as illustrated in Figure 4-4. The system is modular, and readily expandable for addressing systems with hundreds of processors, and well as low end applications.

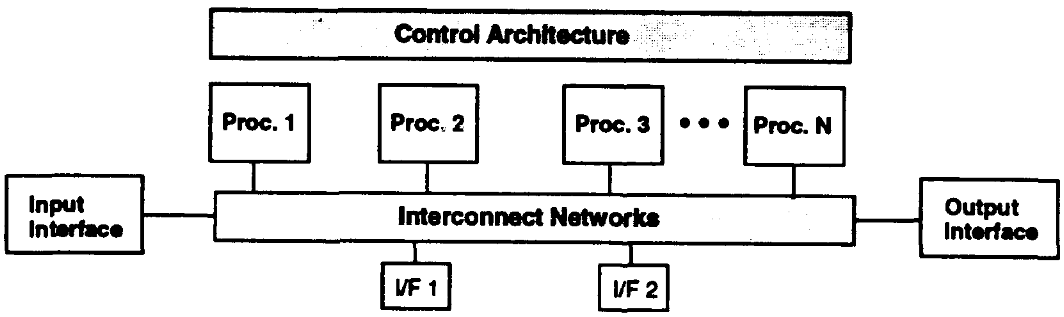


Figure 4-4. Model year architecture concept.

Multiple processors (either homogeneous or heterogeneous types) are accommodated, and selection can be made based on the particular requirements of the algorithm. The processor functions to be accommodated include signal and data processing, and control processing. The signal processing and data processing functions, generally correspond to the nodes of the flow graph. Control processing functions handle coordination of task execution of the processors (initialization, switching functions for multiple modes of operation - ex. multiple flow graphs), control of the network, management of diagnostic functions, and exception processing.

The processors are networked together by a high bandwidth interconnect network (crossbar, bus network, other) for signal data routing. The type and characteristics are determined based on the specific algorithm requirements, although standard interfaces will be employed, unless totally not feasible. Lower bandwidth interconnection is also accommodated for control function support, also implemented with standard interfaces. The characteristics of the various signal processor interfaces which must be supported by RASSP are shown in Figure 4-5. This dictates that the architecture provide a comprehensive set of standardized interfaces. It is evident from Figure 4-6 which summarizes the status of various standards organizations relative to various interconnection approaches, that many of the relevant interfaces are being addressed. Data flow networks and high speed I/O channels, however, are only recently starting to be addressed by the standards organizations, and must receive focused attention on the RASSP program.

Characteristic	Analog Processor	Signal Processor	Data Processor
Function	Signal pre-processing, gain compensation, filtering, etc.	Detection, filtering, CFAR, etc.	Post-processing,scheduling/ control, identification, etc.
Data Unit	Analog (in), Pixel/vector (out)	Vector, pixel, etc.	Words - detections, tracks, regions, etc.
Processing	Analog, conversion	Floating pt. vector	Integer/floating point scalar/vector data
Interfaces			
— Input	Sensor bit stream - 10s Mb/s to Gb/s Control - 10-100 KB/s	Sensor - 100s MB/s Test and maint - Mb/s Control bus - 10s MB/s	Med BW bus/interconnect - 10s MB/S TM bus, serial control - Mb/s
— Output	High BW word stream - 100s MB/s	Data proc. interface- 10s MB/s	Display interface - MB/s Control Bus - 10s MB/s
— Internal	Internal - Low BW control	High BW shared interconnect (100s MB/s per link)	High/Med BW multiproc. inter-conn. (10s-100 MB/s per link)
Memory Reqs.	Minimal	High speed shared buffer	Large shared memory, data base access

Figure 4-5. Current signal processor interface requirements.

The standardized interfaces described above provide support for hardware interoperability; to ensure a truly open architecture, software interoperability must be addressed as well.

Great strides in software interoperability have been made over the past few years withe the acceptance of a number of new interface standards such as POSIX, X/OPEN, and OSF. These have not been applied to a large extent to the signal processing area, mostly due to its embedded nature and the performance impact incurred by implementing these standards. The RASSP software development framework will encompass appropriate standards to ensure software interoperability.

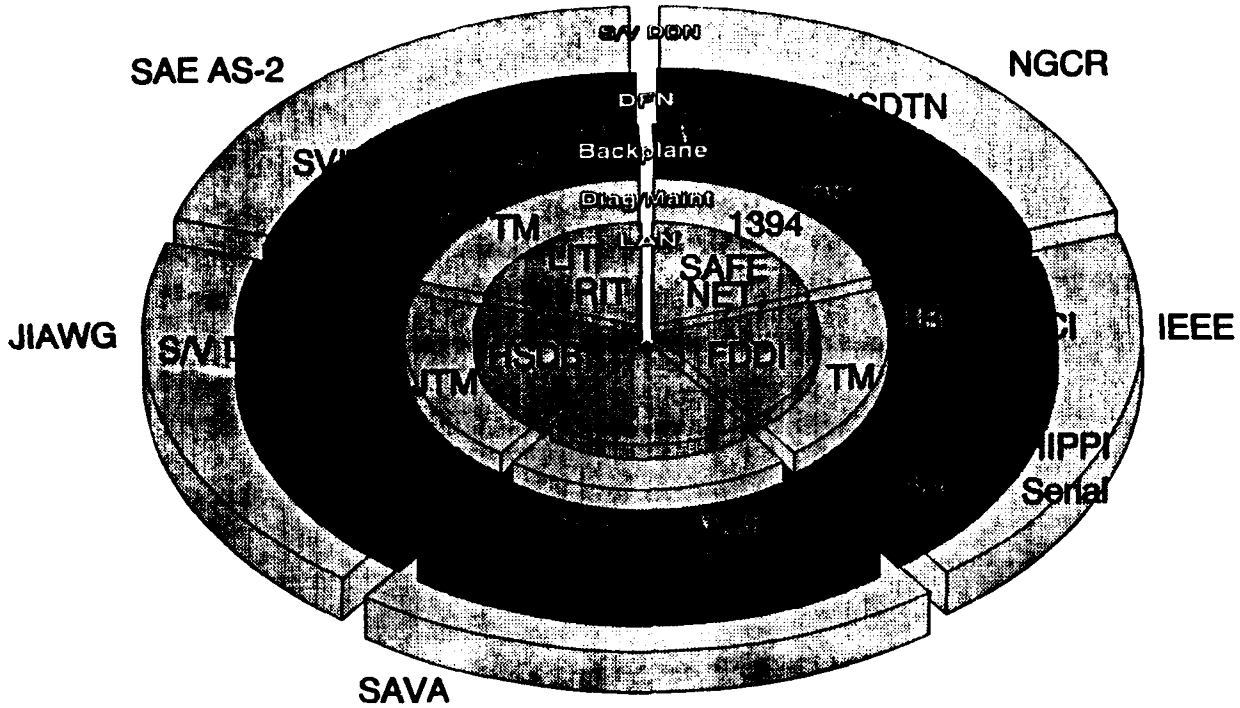


Figure 4-6. Standardization committee status.

Both hardware and software interoperability are greatly enhanced when adherence to ISO/OSI layered architecture standards are employed. For example, as a participant on the Rome Labs Architecture for Survivable System Processing program, GE (working as a member of the Boeing team) developed an open architecture concept that provides for technology-independent interfaces, implementing a "virtual" network throughout the open system to allow for network flexibility and technological evolution.

A specific example of how this is achieved is shown in Figure 4-7. Each module within the architecture includes a common network interface (CNI), which is a multi-chip module that provides the common interface between all processing elements (PEs) and the various interfaces in the system. On the processor side, the CNI provides a generic memory interface to the PEs. On the interface side, transceivers and support logic implement the specific physical network interface, corresponding to ISO/OSI layers 1 (and perhaps) 2. A general purpose processor (with support hardware) controls interaction between these two interfaces, and implements strict adherence in both hardware and software to the ISO/OSI protocols. Software running on the VNI CPU can implement either communications drivers (for dedicated hardware or processors not requiring of OS), or can implement an entire distributed OS microkernal for heterogeneous processing environments.

Using this approach, if a specific interface changes, either to encompass a new standard, or perhaps to upgrade the interface from electronic to fiber-optic technology, the physical interface within the VNI is the only hardware that is modified. In addition, by strict adherence to ISO/OSI standards, the only VNI software which must be changed is the specific interface driver software; the processor support (OS and control) software remains unchanged.

Adherence to such standard interfaces and layered implementations is not without performance penalty. However, as technology improves, this performance penalty is decreasing. In addition, we believe that systems are being driven harder by cost constraints than by performance requirements, and that the majority of future systems will embrace such design techniques. The proposed RASSP methodology will encompass these design techniques to enhance rapid prototyping, design reuse, and technology insertion.

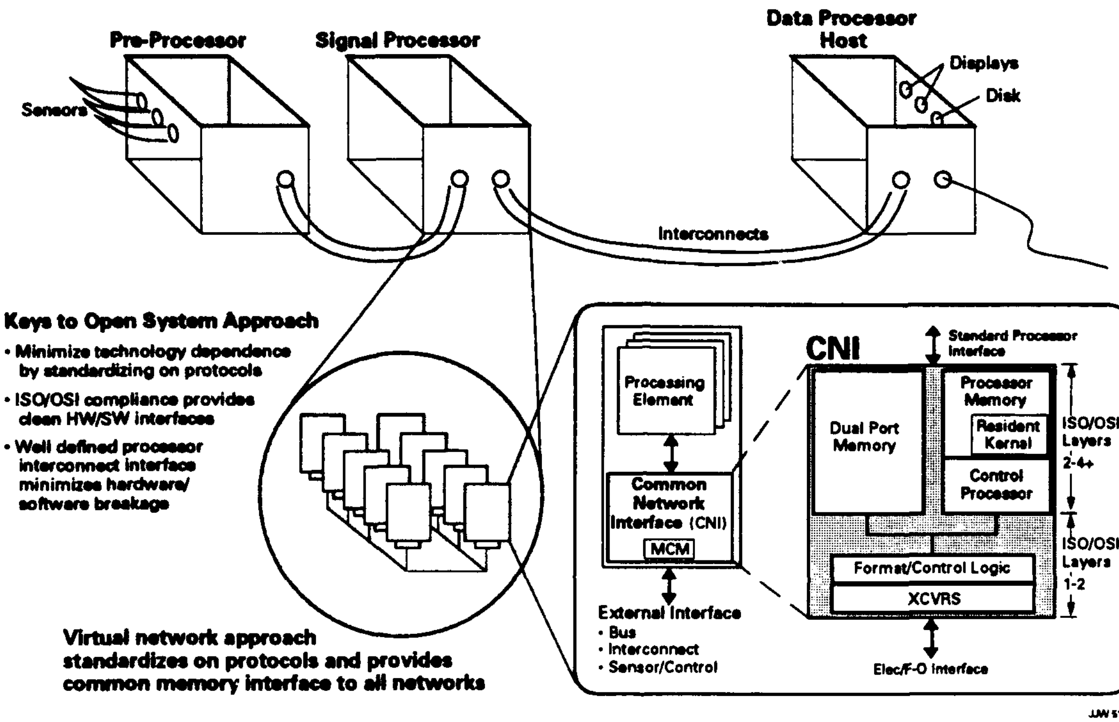


Figure 4-7. Modular open system processing concept.

The RASSP system also supports the Model Year approach to design reuse and upgrade for application and support software. Application code is designed for execution of signal processing algorithms expressed in data flow format. The software is designed by mapping the data flow graphs onto the nodes of the architecture, then automatic generation of the majority of the required code for execution of the nodes on the processors, and for routing of the data (associated with the data flow links) between the processors. The generated code will be HOL, using optimized libraries where relevant. Extensive support for object-oriented programming from signal processing algorithm libraries is thus easily provided. CASE tools are used for documentation generation, and for maintenance of documentation pedigree (supporting reuse strategy).

Operating system and support software will be implemented using standard (POSIX) interfaces, leveraging commercially supplied and supported products. Execution of the algorithms is accomplished by receipt of the input data blocks into the system and parallel operation of the processors and routing network to perform the required functions.

Model Year Implementation Issues: Successful implementation of the model year concept, will require challenges in design methodology, and compromises in design approaches relative to several areas of significance: widespread utilization of open architectures (hardware/software), efficient generation of reusable application software, and technology vendor compliance. Issues related to acceptance of this RASSP technology are shown in Table 4-1.

Req. Technologies	Implementation Issue	The Future — RASSP
• Open architectures — Modular, Open Systems — Standard interfaces	• System reqs. dictate dedicated/custom HW to meet SWAP reqs. • Overhead associated with standards impose penalties • DSP requires minimal communication functionality • Mil standards not compatible with commercial standards	• Cost will dictate use of commercial procs., open system support • Common Network Interface (CNI) isolates functions; performance impact reduced over time • Support range of functions to performance • RASSP will leverage market impact to ensure new standards support requirements
• **Application Software** — HOL-based Retargetable code	• HOL code is less than 50% as efficient as assembly code • Code optimization, new architectures require complete code redo. Mil qual documentation costs	• Better compilers evolving; support optimized macros • Use of CASE, HOL, OO, and DFG tools minimizes retargeting costs; tools should support backward annotation
• **Support Software** — Structured OS — ISO/OSI compliance	• Performance penalty for excess functionality • Efficiency for layered interfaces is too low	• Support range of functionality for ASSPs • Provide flexibility to minimize overhead within ISO/OSI structure
• **Vendor Acceptance**	• Commercial vendors must provide access to sensitive data early • Military vendors must share sensitive data	• Vendors responsible for protected inputs into data base • Cost, schedule benefits of RASSP will ensure usage

Table 4-1. Model Year implementation issues.

Open System Issues relative to RASSP: System requirements dictate development of custom designs to meet size, weight, power, and performance requirements; the overhead associated with standards impose penalties on the designs, communication functionality associated with DSP's is minimal, and does not warrant the overhead of layered models, military standards are not compatible with commercial standards, and the performance penalty associated with standard operating systems is not warranted with signal processors.

Cost will dictate use of commercial processors in lieu of custom designs, the performance impact of software layers associated with standard interfaces will be reduced over time, savings in software development costs through reuse of interface code will outweigh the performance penalty, and RASSP program will leverage market impact to ensure new standards support requirements

Application Software Issues Relative to RASSP: HOL code is up to 50% less efficient than assembly code. Code optimization, or mapping to new architectures requires complete code regeneration. Military documentation costs are high and, therefore, need to be addressed as part of the RASSP design system. The GE-EPI system has started addressing this need and has templates of the 2167 software documents that are supporting the specification capture and documentation generation tools.

Model Year Rationale - Application Software: Better compilers are evolving; support for optimized macros will improve performance. Use of high level CAD tools for autocode generation, and retargetable HOL's will improve software generation cost and schedules associated with the new architectures, the use of CASE tools, and associated back annotation capabilities will minimize documentation effort and cost. The GE RASSP approach is emphasizing software/macro code reuse that will provide significant cost and schedule enhancement.

Vendor Acceptance Issues: Commercial and military processor technology suppliers have been reluctant to supply advanced product information, and models to aerospace designers, because of the competition sensitive nature of the information. However, the recent experience with suppliers like Intel have seen significant improvements. Discussion with Honeywell regarding the Touchstone developments indicates a major step toward the cooperative design of the Paragon/Touchstone program.

Provisions will need to be made within the RASSP data management systems to provide adequate protection for vendor proprietary information, and responsibility for determination of releasable information will be maintained by the vendors. Utilization of RASSP will become sufficiently widespread, that a financial incentive will exist for vendors to supply the necessary produce advanced release information, and models.

The GE RASSP concept will provide models that can be encoded or protected by other means for use in the design of processors for DoD applications. Approaches like the Zycad CAD model bank is an approach being considered.

GE understands that CFI, Logic Modeling, Intel, T.I., Motorola and others met recently to discuss how to allow early release of models through encoded or other release concepts. Based on this ongoing discussion, GE believes it will be possible to evolve over the course of the Phase II program an approach that will permit early release of new design models.

5. DESIGN METHODOLOGY AND DESIGN SYSTEM REQUIREMENTS

5.1 System Design Methodology

The RASSP system design methodology features a top-down hierarchical approach shown in Figure 5-1. In the system requirements process, top level system level concepts are developed and tradeoffs are performed to define the subsystem requirements. Emphasis is placed on the RASSP program on the design and development of the signal processing subsystem. The three key components of the signal processor design are: 1) algorithm definition and validation during the subsystem requirements process, 2) processor architecture definition and algorithm partitioning/mapping during the architecture development process, and 3) concurrent hardware and software development. The signal processor design is electronically linked to an automated manufacturing facility in the RASSP system. A key feature in this design methodology is the joint analysis, simulation and construction of the signal processor which provides for ease in integration and timely design feedback for rapid prototyping. In this section of the report, the required tasks, design examples, CAD tool requirements, available CAD tools and developments required for RASSP are discussed for each process of the design methodology.

5.1.1 Required Tasks/Functions

The system definition process is a front-end system engineering activity in which system level concepts are developed to meet customer requirements and top level tradeoffs are performed to define the subsystem requirements of the system. As shown in Figure 5-2, the system definition process is composed of three tasks: requirements analysis, functional decomposition, and functional allocation. Each of these three tasks are described below.

System Requirement Analysis: The mission and procurement requirements are initially examined in this task to ensure that all requirements are well understood. There is close interaction with the customer during this task to clarify any confusion with the system requirements. The system requirements are electronically captured so that a traceable path can be established when the requirements are allocated to functions and components. Both mission and threat analyses are performed to understand how the system should behave. The system is defined by describing the system modes, functions and interfaces. Measures of effectiveness (cost, performance, risk, etc.) are established for the system to provide metrics to compare different system architectures. Operational scenarios are developed which will be used to determine the system performance.

Functional Decomposition: The system is decomposed into its functional elements after the system requirements have been established. This functional decomposition is performed by determining what functions are required to implement each system requirement. Functions are described by defining the inputs to the function, the algorithm performed by the function and the outputs of the function. Constraints and timing requirements for each function are identified. Waveforms are defined for each of the operational modes of the system during the functional decomposition task. The top level system behavior is modeled to determine the functional performance of the

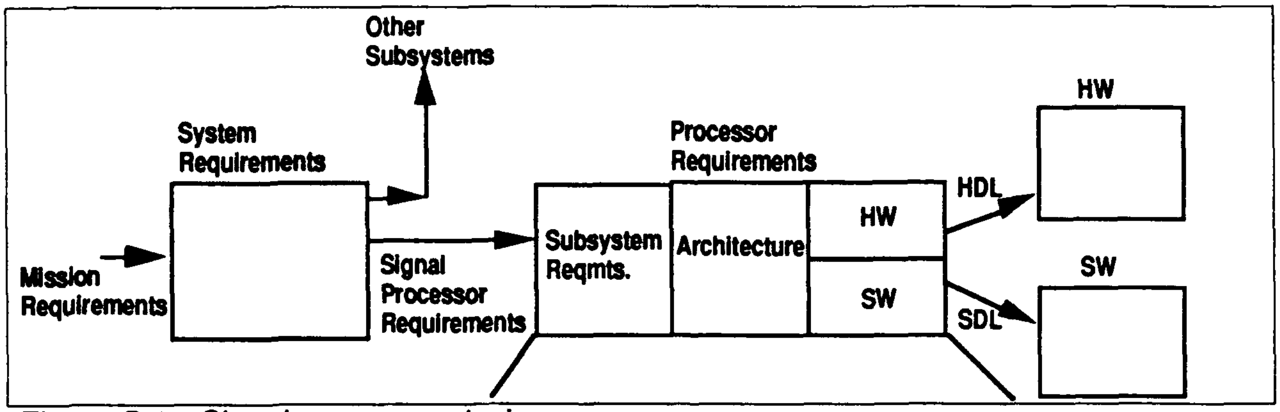


Figure 5-1. Signal processor design process.

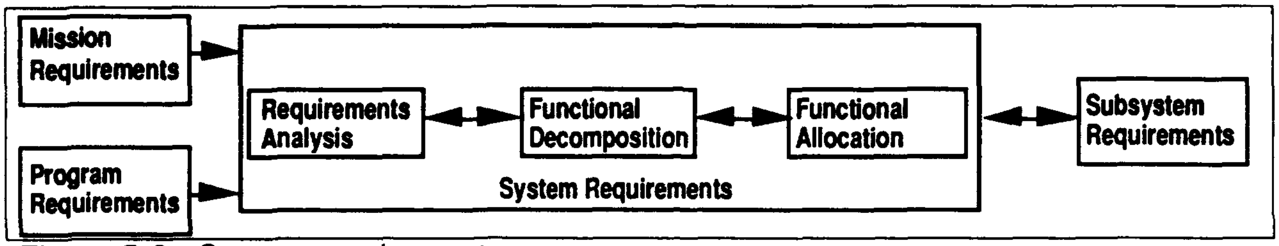


Figure 5-2. System requirements process.

system. Signal-to-noise ratios, detection ranges, probability of detection and probability of false alarms are several examples of system level behavior for a detection system (radar, sonar, etc.). System test and maintenance concepts are developed for this task. All functions within the system must be traced back to the system requirements.

Functional Allocation: The functions of the system are allocated to subsystems once the requirements have been established. At this point various system architectures are developed and characterized to determine the baseline system. Tradeoff analyses are performed to determine the most viable architectures. Tradeoff analyses are typically performed for the following areas: reliability, availability and maintainability; life cycle cost; schedule and technical risk; integrated logistics support; human factors; and system safety. All system requirements must be traceable to both functions and subsystems.

The output of the system definition process is a set of requirements for each subsystem. These requirements include system mode descriptions (search, track, waveforms, etc.), performance requirements (processing gain, noise level, detection range), subsystem constraints (size, weight, power. cost), and interface requirements.

The system definition process is an iter- , which requires significant interaction with both the customer and subsystem designers. Automated links must be provided between the system and subsystem designers so that the impact of lower level design detail can be assessed directly in system level performance simulations.

5.1.2 Design Example

GE's Advanced Technology Laboratories (ATL) was recently the prime contractor for the Air Force (Rome Air Development Center) for the concept development of a VHSIC signal processor for the Joint Surveillance and Target Attack Radar (JSTARS). JSTARS is an airborne radar system that monitors troop movements, classifies targets and directs weapon systems. The focus of the GE contract was the concept development of a VHSIC signal processor which would provide a two to one reduction in the size and volume of the signal processor under development for JSTARS. The VHSIC signal processor would provide improved functional performance, have better reliability and provide more performance margin than the baseline processor. The work performed on GE's JSTARS contract provides a good set of design examples to illustrate the system design methodology.

The first step in the system definition process is to analyze the mission and procurement requirements. For JSTARS there are four primary missions supported: surveillance and threat analysis; attack planning; attack control; and post attack assessment. The operational and radar execution modes to support these four missions are shown in Figure 5-3. Each mode is described in Table 5-1. The priority for each mode shown in this table has been encoded for classification reasons.

After the modes have been defined, the radar surveillance volume, waveforms, processing functions, key radar parameters and requirements must be defined for each mode as shown in Table 5-2. Radar performance simulations must be executed for each mode to ensure that sufficient performance (detection range, signal-to-noise ratio, image quality, etc.) is obtained.

The functional requirements are then allocated to subsystems. A functional allocation of the processing functions for the MTI mode for the JSTARS system is shown in Figure 5-4. Each of the functions must be defined in terms of its inputs, outputs and processing gain.

5.1.3 CAD Tool Requirements

Five classes of system definition tools are needed for RASSP. These tools include: requirements traceability support, functional simulation support, tradeoff analysis (functional partitioning) support, life cycle support, and document support. The requirements for each of these tools are discussed below.

Requirements Traceability Support: System requirements should be captured in an electronic data base for easy access. Relationships between the system requirements, derived requirements, functional decomposition and allocation should be maintain within the data base. The requirements traceability tool should maintain a traceable path for all critical issues and decisions during the entire system development. The tool should check consistency to ensure that all requirements have been allocated and that all system functions, components and interfaces are completely linked. The requirements traceability tool should maintain configuration management for the system requirements and baseline architectures. The tool should be able to quickly determine the impact of requirement changes on the system.

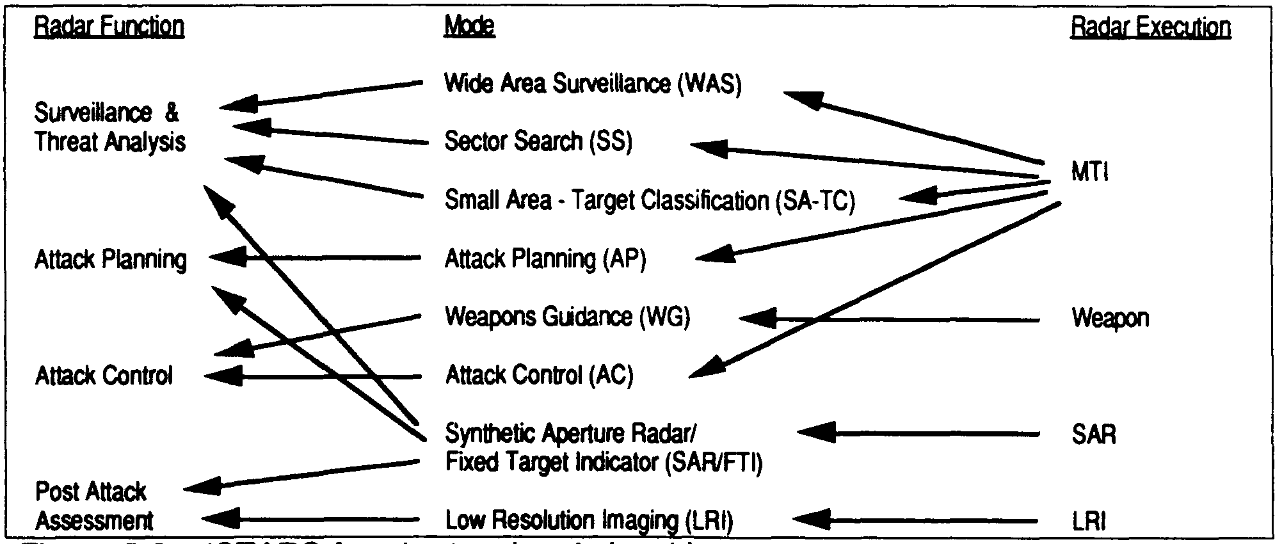


Figure 5-3. JSTARS function/mode relationships.

Table 5-1. JSTARS mode description.

Mode	Priority	Purpose	Description
WG	N1	Midcourse weapon guidance	• Revisit times • Communication bandwidth
AC	N2	Medium resolution search of small areas	• Update period
SAR/FTI	N3	SAR imagery or fixed target indicator	• Dwell time • Latency time
SA-TC	N4	Medium resolution surveillance of single azimuth beam	• Latency time
AP	N5	Medium resolution search of small areas	• Revisit time
SS	N6	Low/medium resolution surveillance	• Revisit time
WAS	N7	Low resolution surveillance of wide areas	• Revisit time
LRI	N8	Low resolution imaging of large areas	• Resolution • Surveillance volume

Table 5-2. JSTARS functional requirement summary.

Function	Mode	Coverage	Waveform	Processing Functions	Radar Product	Key Requirements
MTI	AC SA-TC AP SS WAS WAS-TC	• Surveillance area	• Waveform description	• Pulse compress. • Motion compensation • Doppler filtering • CFAR • Target classification	• Target range • Range rate • RCS • Classification	• Accuracy • Pd vs range • Update rate
SAR/FTI	SAR FTI	• Surveillance volume	• Waveform description	• Pulse compress. • Motion compensation • Cross range processing	• SAR image	• Contrast ratio • Resolution parameters • Pd
Weapon	WG	• Surveillance volume	• Waveform description	• Pulse compress. • BIT location • BIT detection • Message decode • Weapon update	• Weapon midcourse guidance	• Acquisition probability • Update rate

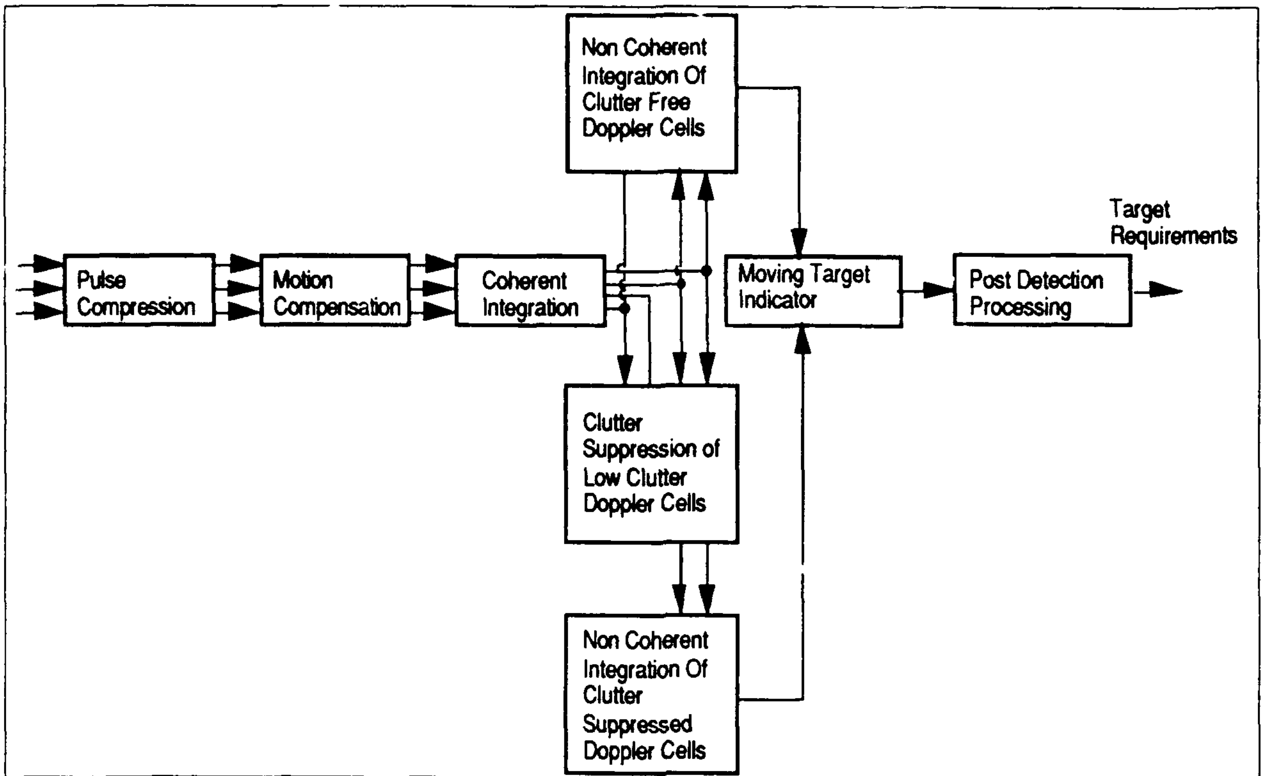


Figure 5-4. JSTARS signal processing functions for the MTI mode.

Functional Simulation Support: The functional simulation should support both basic types of performance simulations: functional performance simulations where system level parameters such as signal-to-noise ratios and detection ranges are determined and timeline simulations where system mode control and timing analysis are performed. The simulation tool should contain both a natural mathematical equation and functional block diagram entry capability. This tool should be compatible with lower level design tools so that data can easily be passed between simulators. A wide variety of data visualization techniques should be a part of the tool. For configuration management the tool should maintain a data base of simulation results with links to files containing the input parameters used.

Tradeoff Analysis (Functional Partitioning) Support: The tradeoff analysis support tool should determine key system metrics such as cost, risk, schedule and size for alternative system architectures. This tool should support libraries from previous system designs. The tool should contain a design advisor to assist the system engineer in performing the functional partitioning of the system.

Life Cycle Support: The life cycle support tool should contain a data base of past system designs which can be used as inputs to perform failure analysis, reliability calculations, and life cycle costs. Links to cost estimation tools (like the GE Price System) provide early estimation capabilities to drive analyses.

Documentation Support: The documentation support tool should provide the capability to provide different views of the requirements data base and provide templates for the major military standard reports.

5.1.4 Summary of Available Tools

The GE team has surveyed many vendor tools during phase 1 of the RASSP program. The data for the survey was collected from vendor demonstrations and product brochures. GE's understanding of the capabilities of current tools supporting the system definition process is summarized in Table 5-3. As shown in this figure, these tools can be grouped into four main categories: requirements traceability, mathematical analysis, functional performance/timeline analysis and life cycle support.

Table 5-3. Capabilities of existing system design tools.

	Requirements Traceability		Mathematical Analysis		Functional Performance/Timeline							___ Cycle	
Requirement	Ascent Logic RDD-100	Marconi RTM	Wolfram Mathematica	Math Works MATLAB	Mentor Graphics SDS	Mentor Graphics DSP Station	COMDISCO SPW	COMDISCO Bones	SES Workbench	Nuthena Foresight	I-Logic Statemate	GE PRICE	DAZIX FFAP, MEAP & Cheap
Requirements Traceability													
• Consistency Checks	C	C			P								
• Simulation Support			L	L	P		C	C	C	C	C		
• Configuration Management	L	L									C		
Functional Simulation Support													
• Mathematical Analysis			C	C	P	L	L						
• Timeline/Performance	L	L			P	L	L	C	C	C	C		
• Hierarchical Link to Tools					P			C	L	C			
Tradeoff Analysis Support													
• Manual			L		P							L	
• Design Advisors													
Life Cycle Support													
• Cost Modeling												C	C
• Logistics Support													
• Reliability/Maintainability													C
Documentation Support													
• Generation/Maintenance	C	C											

Legend: C = Has Capability
P = Planned
L = Has Limited Capability

For the requirements traceability tools, Ascent Logic's RDD-100 and Marconi's RTM provide the greatest capability. RDD-100 was chosen as the primary GE Aerospace requirements traceability tool for the internal Engineering Process Improvement (EPI) program. The requirements traceability tool that is selected for RASSP should be linked to a system simulator.

A wide variety of vendor tools satisfy the functional simulation support tool. Wolfram's Mathematica and Math Works Matlab provide the most capability for mathematical analysis. Mentor Graphics' System Design Station (SDS) is under development in Europe to provide a high level system design tool. Mentor Graphics' DSP station and Comdisco SPW provide a functional performance block diagram analysis tool. Comdisco's Bones, SES Workbench and I-Logic's Statemate provide a timeline analysis/network simulator tool. Nuthena's Foresight combines the functional simulator and timeline analysis tool within the same tool.

There are no apparent tradeoff analysis tools to assist in performing the functional partitioning of a system.

Dazix's Reap, Meap & Cheap tools provide reliability, maintainability and cost modeling support. GE's Price tools also provide life cycle cost modeling support.

The requirement traceability tools, RDD-100 and RTM, provide documentation support.

While these tools provide many of the required capabilities, none are well integrated to each other, nor do they support hierarchical tool links to support easy integration.

5.1.5 <u>Required Developments</u>

A wide set of integrated tools whose capabilities span from top level system design to electronic links to manufacturing are required for the RASSP program. Resources are far too great to attempt to simultaneously develop each of the RASSP tools. In addition, technology developments are needed to develop certain parts of the RASSP tool set. One of the greatest challenges on the RASSP program is to select those areas for development which will provide the greatest impact in designing new processing systems. The following areas are recommended for development to support system design under the RASSP program.

<u>Requirements Traceability:</u> Integrate the requirement traceability tool with the selected RASSP framework. This task will link the requirement traceability tool with the functional performance and timeline simulators.

<u>Functional Simulation Support:</u> Many scenario generators and system level simulators have been developed within the government and industry. The intent on the RASSP program is not to redevelop these programs using a common system simulator. The main task proposed for development on the RASSP program is to link the mathematic analysis tools (Mathematica/Matlab) with the selected functional performance and timeline simulators. Note that the development of the functional performance and timeline simulators is an important part of the signal processor system design and improvements in these simulators are discussed in the signal processing section of this report.

<u>Tradeoff Analysis Support:</u> Tradeoff analyses are required on the system level when performing the functional partitioning of a system. A system level design assistant which would assess the cost, risk, schedule,size and other system metrics would be of great use to the system designer. However, it is felt that the technology needed to develop this type of design assistant is not very mature. Design assistants need to be developed for system components such as signal processors before they can be developed at the system level. No development activities for system level tradeoff analysis tools are recommended for the second phase of the RASSP program.

Life Cycle Support Tools: Integrate the GE Price tools into the selected framework to provide a life cycle cost modeling capability for the RASSP system.

Documentation Support: Integrate the selected tool set with a standard desktop publication package such as Interleaf.

5.2 Subsystem Design

The signal processor subsystem design is based on requirements inherited from the overall system definition. These requirements consist of the signal processor's modes and function, its interfaces to the other systems, and the signal processor system constraints. The signal processor system modes and functions may be for example: wide area surveillance, search, and track. The functions often include the basic equations to be implemented. The interfaces to other systems are for example: sensor elements inputs, control inputs, and displays outputs. Examples of signal processor system constraints are size, weight, power, and cost. The overall system design is based upon statistical and probabilistic analysis and knowledge of existing capabilities to specify and meet the requirements such as: detection range, PD, and PFA as covered in Section 5.1.

The signal processor system design processes described below are intended for systems containing multiple interconnected processing units. The processing units may differ in function and design. Some units may be designated for control and others may be reserved exclusively for signal processing. Alternatively, control and signal processing functions may coexist within processors. The processing system may be parallel or distributed, SIMD or MIMD, or it may contain combinations of these.

The signal processor subsystem design process consists of three major stages: signal processor subsystem concept and algorithm definition, architecture definition, and detailed hardware/software specification and implementation specification. These processes are described in Sections 5.2 through 5.5, respectively.

5.2.1 Subsystem Concept/Algorithm Definition

The signal processor system concept and algorithm definition phase is concerned with implementing the required functions/equations on realizable hardware. It does not involve any notion of a processing architecture, but it does develop an algorithmic implementation of the required functions in the form of a pure Data Flow Graph. The operational precision requirements are determined in this design phase through sensitivity analysis and "bit-true" design. More detailed representations of the functions/equations are expressed from operational primitives and library functions. The inter-relationship and data dependence among operations is expressed for accomplishing each function. The underlying control flow is also developed. Tools such as mathematical analyzers, bit-true simulators, and requirements documenters/consistency verifiers are used in this stage of the design process.

5.2.1.1 Required Tasks/Functions

The system concept/algorithm definition task is divided into five general activities as shown in Figure 5-5, Test Generation, Mathematical Analysis, Algorithm Simulation, Algorithm Selection, and Data Flow Graph Generation. Each of these tasks are described below.

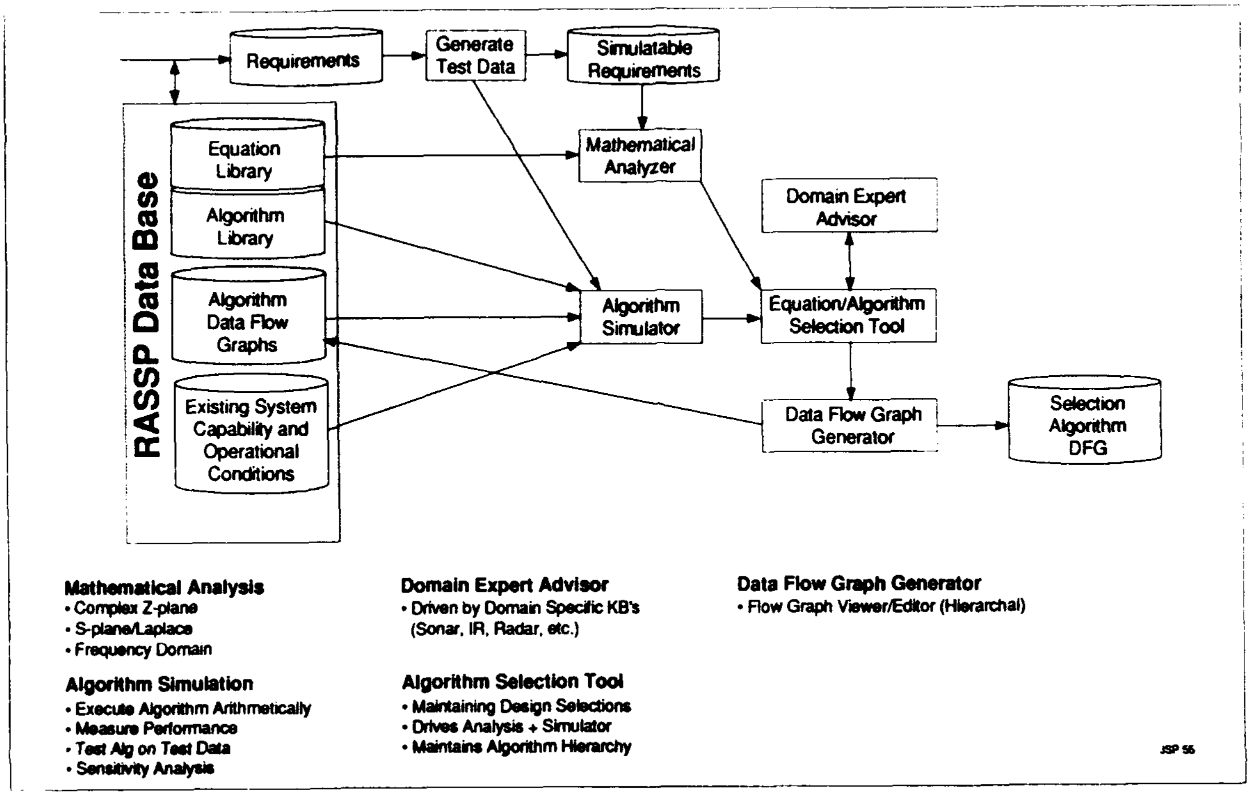


Figure 5-5. Subsystem requirements analysis process.

Test Generation: Test Generation is the translation of the system requirements and specifications into simulatable tests. The tests are composed of stimulus patterns paired with expected output results. Such tests can be used at all design stages, and can be re-run whenever design changes are made to provide automatic verification that the design satisfies requirements. The test data are used to drive the algorithm simulations and evaluate the results.

The test generation process also provides a means for analyzing, understanding, and checking the completeness and consistency of the requirements, since the requirements must be thoroughly understood to generate such tests. The tests are also used by the other tasks. Therefore, the test generation should be the first task performed in the signal processor definition process.

Mathematical Analysis: The Mathematical Analysis is used in the selection/ development of the signal processing equations and algorithms. It provides for the analysis of systems in the complex Z-plane, Laplace S-plane, and frequency domains for comprehensive understanding of system performance and design. Mathematical analysis tools are also used to synthesize components such as filters and their coefficients. Under analysis, the basic equations and algorithms are developed and verified for arithmetic correctness.

For efficiency, usually the quantities manipulated under analysis are statistical or representative of real quantities, but not the data itself. For instance, such quantities include: S/N-ratios, gains, rejection ratios, mean energy, variance, power, condition numbers, eigen value ranges, and signal and filter transforms.

Tools for this task should operate with and provide natural input, manipulation, and display of mathematical symbols and equations. They should operate and provide translation into the analytical domains with appropriate and flexible visualization/ graphing capabilities. They should inter-operate with the tools used in the related tasks through data flow graph interfaces and common data formats.

<u>Algorithm Selection:</u> Algorithm selection is the development of a suitable set of algorithms that satisfy all the relevant requirements. The process usually starts with the development or specification of the underlying equations. The equations are developed and verified through mathematical analysis. Next, practical algorithmic implementations of the equations are developed. The algorithms may be assembled from libraries of known routines, or they may be specified directly. This entails specifying the structure of the overall algorithm in terms of a sequence of filters, thresholds, transforms, mixers, compressors, convolvers, detectors, and etc. For each of these components, selection of fixed or adaptive elements, and the particular algorithm implementation is chosen.

Requirement satisfaction in terms of the targeted performance and computational efficiency are then tested or verified by simulations of the algorithms with test data. The simulation can be used to provide indications about the sensitivity of the algorithm to numerical effects from noise, precision, quantization, sampling, and etc. This overall process is usually iterative. Feedback provided by the simulation is used to modify the algorithms which are then re-tested. The process continues until the specifications are satisfied. The output of this process is the algorithms, the data flow graphs, the control flow, and the precision requirements.

<u>Algorithm Simulation:</u> The algorithm simulation executes prospective algorithms on real or simulated test data. The purpose of algorithm simulation is to investigate numerical properties that cannot be easily studied or understood through analytical means. The sensitivities to various hardware limitations are determined, such as, dynamic range and precision limitations, round-off, quantization, distortion, sample jitter effects, etc.

The algorithm simulation helps provide additional understanding and insight into algorithm performance. It allows interaction with the algorithms at an early time, and it provides a rapid prototyping capability for the candidate algorithms and software. Algorithm simulation facilities should provide ext[illegible]e visualization facilities to visualize algorithm effects on data. Such simulat[illegible]s provide the ability for quickly manipulating algorithm parameters in addition to rapid feedback on modifications and experiments on algorithm variations.

Algorithmic simulation facilities must be linked with their related analytical and modeling development tools. They should accept test data formats from the test generator and facilitate the automatic requirements checking inherent in the tests.

They should also accept algorithm formats as generated by analysis and algorithm selection tools. Simulations should also accept algorithm formats in the form of DFGs, code from auto-code generators, and they should provide graphical block diagram input interfaces for quickly configuring and modifying algorithms.

Domain Specific Expert Advisor: Knowledge based design advisors assist in automating the development process by guiding the designer in selecting and developing algorithms. The knowledge bases are specialized toward specific applications such as radar, sonar, IR, and communications. If the design advisors are coupled to the algorithm development tools as an integral part of the process, then they can produce rapid feedback on modifying the design to meet the requirements.

For example, typical design advisor activities during the design of a radar signal processor might be as follows: suppose a task for the processing system is pulse compression. Then based on the sample rate and pulse length, the design advisor could suggest one of several methods such as tapped delay line, saw filter, FIR filter, or convolution in the time or frequency domain.

An advisor can also help in quickly evaluating the alternatives based on the current requirements, tentative system parameters, and data in the knowledge base from existing designs. A process management tool which oversees and guides the development process is needed to efficiently evaluate the tradeoff alternatives.

More extensive discussions of design advisor technology for RASSP is in Section 6.2.

5.2.1.2 CAD Tool Requirements

Five classes of tools are needed for the RASSP signal processor concept and algorithm definition. These tools include the test generator, algorithm selector, domain specific expert advisor, mathematical analyzer, and algorithm simulator. The requirements for each of these tools are described below.

Test Generation Tools: The test generation tool must accept the formal sub-system requirements and aid in decomposing them into a set of tests which concisely check for system's satisfaction of these requirement's. The test generation tool must provide capability to synthesize simulated test signals. The generation facility must provide the flexibility to specify signal structures of arbitrary complexity. A general mathematical symbolic interface is desirable. Although, a programmable interface is sufficient. Additionally, means to incorporate real data into the test signals should be provided.

The test generator must provide means to pair expected results with the tests. To do this, it should have the mathematical or programmable capabilities described above, and/or easily accept compatible data from the other modeling tools such as the mathematical analyzer. The test generator must produce output test format which is compatible to the related analysis and simulation tools.

The test generator should make use of standard formats for conveying all types of requirements data, not only electrical/behavioral/ logic, but also physical/mechanical.

The test generator should check for requirements consistency and completeness in the generated tests.

Mathematical Analysis: The mathematical analyzer should allow natural mathematical equation entry and interpretation in the form of standard mathematical symbols, expression, and formats. It should provide versatile data visualization functions for viewing the operation, and performance of algorithms and the associated data. The data generated by mathematical analysis such as synthesized signals and filter coefficients, should be transmitted in formats that are compatible with- and acceptable by- the other modeling tools. The normal mode of use should permit and encourage the integration of documentation with the driving equations. The equation format should allow extraction to the algorithm simulator and auto-code generation tools. The analyzer should allow concurrent and interactive execution with other modeling tools, especially the lower level simulations, to provide mixed-level simulation.

The mathematical analyzer should facilitate requirements verification by acceptance and utilization of test and expected result data. The tool should inform the designer of test acceptance status. It would be desirable for the mathematical analysis tool to possess an intelligent interface to the domain specific knowledge-base for automatic access and inclusion of available equations and algorithms.

Algorithm Selection: The algorithm selection tool must be capable of interfacing to and invoking the mathematical analyzer, the algorithm simulator, and the expert advisor. In response to the advisor's suggestions, it should invoke the analyzer and simulator on prospective algorithms. Most importantly, the algorithm selection tool should maintain the algorithm hierarchy, and the history of design selections, experiments, and outcomes as the development proceeds.

Algorithm Simulation: The algorithm simulation tool should accept description of algorithms in the form of hierarchical Data Flow Graphs (DFG) and High Level Language (HLL) code. The lowest level of the DFG hierarchy should be HLL code. It would also be desirable to accept mathematical equations. The algorithm simulation tool should provide means to specify and inject various numerical effects such as, quantization, round-off, precision, dynamic range, distortion, and thermal noise. It should provide means of assessing the algorithm performance in the context of the test data and embedded expected results data. The tool should inform the designer of test acceptance status. The tool should accept data formats from the test generator and mathematical analyzer. The simulation tool should provide versatile data visualization functions for viewing the operation, and performance of algorithms and their associated data.

Domain Specific Expert Advisor: The design advisor must produce rapid feedback on modifying the design to meet the requirements. It should accept an understanding of the required goals. It should be optimized for specific applications by way of domain specific knowledge bases. The design advisor should accept the relevant analyzer and simulator results so that it can rapidly assess the current design status and deficiencies relative to the requirements. The design advisor should suggest and evaluate alternatives based on the current requirements, tentative system parameters, and data in the knowledge base from existing designs.

5.2.1.3 Summary of Available Tools

Many vendor tools were surveyed during the first phase of the RASSP program. Data on the tools were collected from vendor demonstrations and product brochures. Time did not permit full evaluations of each product. Table 5-4 summarizes GE's understanding of the current tools which support the signal processor system concept and algorithm development.

Table 5-4. Summary of available architecture tools.

Requirements	SES Workbench	Foresight Nuthena	Comdisco Bones	Comdisco SPW	Mentor DSP	ILogic Statemate	Micon CMU/ Omniview
Algorithm Development			C	C	C		
DFG Interface							
- Hierarchical	C	C	C	C	C	C	
- Versatile Visual.		C	C	C	C	C	
Design Verif.							
- Algorithm		L	C	C	C		
- Sensitivity			C	C	C		
- Link to Arch Tools			C		L	C	C
Hierarchical Tool Links							
- System Tools			C	P			
- Arch Tools	C	C		L	C	C	C
Documentation Support							

Legend:
C = Has Capability
P = Planned
L = Limited Capability

5.2.1.4 Required Developments

Emphasis is placed on capabilities that are unlikely to be developed by industry in a timeframe compatible with RASSP, however will produce the greatest improvement for RASSP. The following areas are recommended for development under the RASSP program.

Test Generation: There are many test generators at the logic level, but test generation at the system concept/algorithm level appears to be ad hoc. There are several requirements analysis tools. Though none are fully connected to an automatic design checker. There are currently no standards for embedding all requirements data (including physical and mechanical) into such automatic tests.

Most of the components for a test generator as described in Section 5.2.1.1 exist and have been demonstrated, such as requirements analyzers and mathematical analyzers. Some critical aspects require development, such as a means for encoding the physical and mechanical information. A task which integrates these components and develops the necessary extensions would have high payoff for RASSP because so may other tools depend on this automatic requirements testing capability.

Mathematical Analysis: Several good tools exist for mathematical analysis. However, none are tightly integrated into the signal processor design environment. For instance, couplings to data flow graph algorithm description and auto-code generation formats and tools would greatly accelerate the usage of such tools for signal processing system development in the RASSP environment. A task is proposed to select a mathematical tool and develop methods to integrate it into the RASSP environment by implementing the described interfaces.

Algorithm Selection: There are several tools available which exhibit many of the features required by the algorithm selection tool. Many of the constituent components such as data bases, framework tools, and version managers exist or have been demonstrated. Therefore, GE recommends the selection of an existing tool as a basis, with additional incremental development to implement the needed features. The development should be based primarily on integrating existing components.

Algorithm Simulation: Some algorithm level simulators exist. However none appear to offer all the features required by RASSP. Therefore, this task would select the most appropriate simulation tool and incrementally develop the extended capabilities. Such capabilities may be for instance, compatibility to the other tools, enhanced visualization, and hierarchical mixed-mode co-simulation.

Domain Specific Expert Advisor: No design advisors are known to exist for algorithm and system concept development. Advisors have been demonstrated, with excellent payoff, in similar application areas. Therefore, a task is recommended to select a design advisor system, and create a domain specific algorithm selection knowledge base. Later, a similar knowledge base for signal processor system concept development would be generated.

5.3 Architecture Definition

The architecture definition phase is divided into four basic tasks as shown in Figure 5-6, partitioning/mapping, hardware/software codesign, architecture selection, and architecture/data flow simulation. Each of these tasks is described in the following sections (5.3.1 to 5.3.5). In addition, Section 5.3.6 describes Hardware Description Languages (HDL) and extensions which are required to meet RASSP goals.

5.3.1 Architecture Partitioning/Mapping

In the partitioning phase, tasks are partitioned between analog hardware, digital hardware, and software. The tasks are further partitioned among the components within each of these categories. For instance, on the digital hardware side, the processing hardware is assigned various tasks such as control, I/O formatting, and processing. On the software side, the software structure is partitioned according to sub-tasks.

In the mapping phase, nodes of the Data Flow Graph (as defined in the system concept/algorithm definition task) are assigned to the individual processor elements within the candidate architecture. These nodes consist of operations to be performed on the data that are represented by the arcs of the data flow graph. In addition to the operations, the data must also be mapped to locations within a parallel architecture. For instance, at one stage data may be distributed among the processor element's

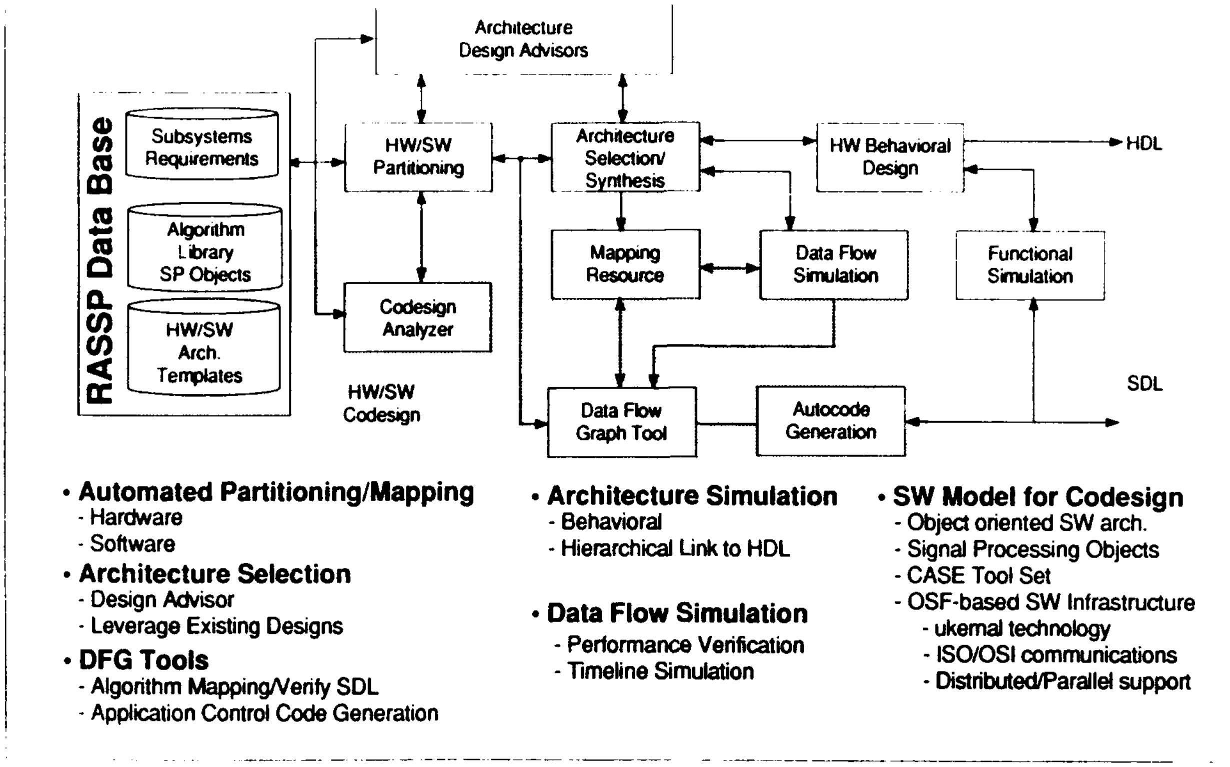


Figure 5-6. Architecture development process.

local memories, while other data may be stored in centralized shared memories. The data location changes with time as operands are supplied to the computation units.

To initiate the partitioning process, an algorithm designer transforms the system requirements into a Signal Flow Graph and specifications on the performance of nodes and paths in the Signal Flow Graphs. Signal Flow Graphs are composed of high functionality nodes (such as Fast Fourier Transforms) and arcs representing block data transfers. The System architecture and Hardware/Software Partitioner analyzes the Signal Flow Graph and the Performance Specifications producing an initial hardware and software architecture and a mapping of the Signal Flow Graph into multiple hardware and software partitions. The hardware is synthesized by a hardware synthesis system. A Software Codesigner (described in Section 5.3.2) generates the software modules to be run on the synthesized hardware. Feedback from the Hardware Synthesizer and Software Codesigner to the System Architecture and Hardware/Software Partitioner produce a new set of hardware and software specifications which in turn can be synthesized, producing a superior design. The system iterates until convergence is reached.

5.3.1.1 Functional Partitioning/Mapping

The System Architecture and Hardware/Software Partitioner (SA&HSP) is depicted in Figure 5-7. The SA&HSP receives a Signal Flow Graph and a set of performance specifications as input. A typical Signal Flow Graph from a Navy application is depicted in Figure 5-8. The nodes in the graph represent high-level signal processing primitives such as Fast Fourier Transforms, low pass filters, weighing functions, and detection functions. Associated with each node is the number of operations and associated with each arc is the amount of data which must be transferred from one node to the next. For example, Node 2 in Figure 5-8 is a Fast Fourier Transform which receives 16,000 complex numbers as input and produces 16,000 complex numbers as output. The transformation requires 20,000 operations. Similarly, Node 4 is a detection algorithm which takes 16,000 complex numbers as input and produces 8,000 real numbers as output requiring 10,000 operations. The whole graph from input to output must be performed in one second.

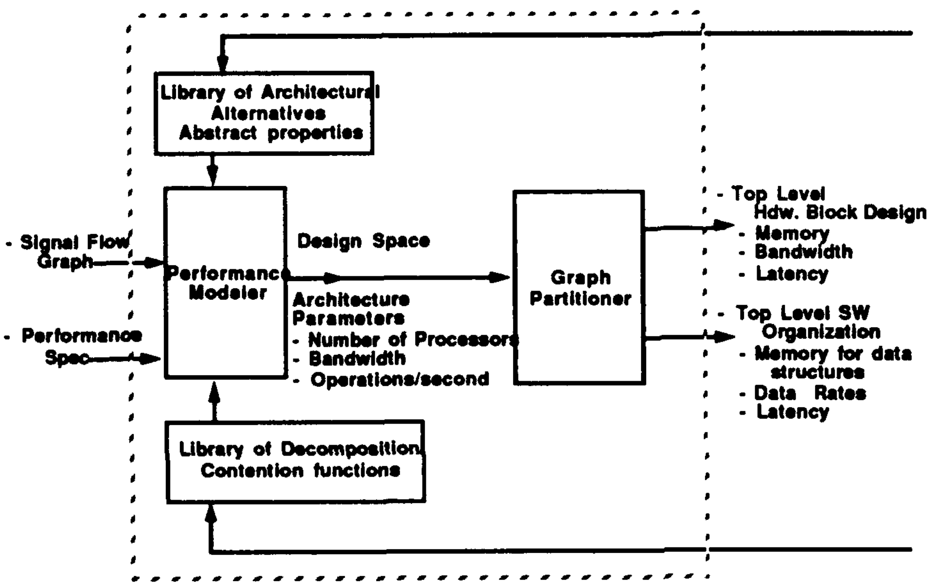


Figure 5-7. System architecture and hardware/software partitioner.

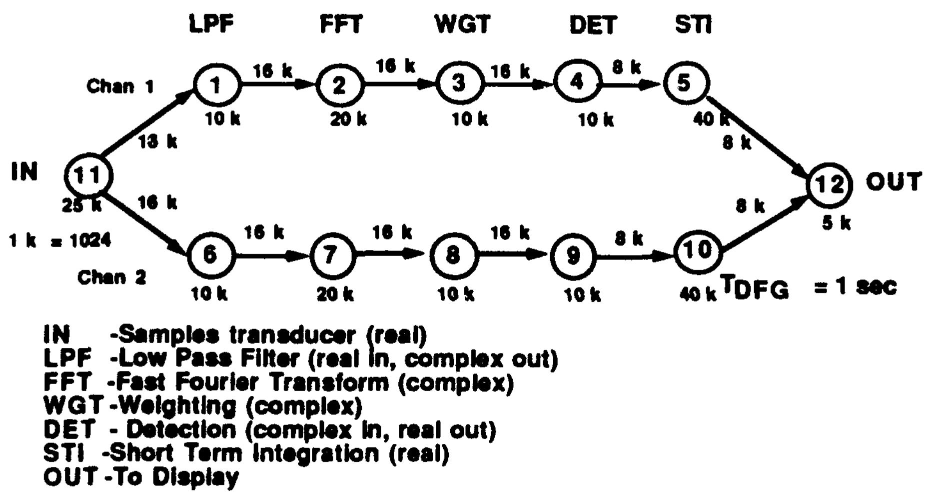


Figure 5-8. Two-channel CR Lofar signal flow graph.

To efficiently model the performance of the DFG, several important factors must be known or estimated. One key item in representation of the DFG execution times is the code execution time. In general, the code modules that compute the signal processing algorithms are autocode generated and timing can be established by code execution directly on the candidate architecture or simulator or by estimation based upon equivalent C code run times of the host machine. The other source of timing data is the inter DSP data and communications times. In order to facilitate code development with interprocessor communication, a real time executive code library is needed. These modules are well characterized by parameters for each subroutine/process and therefore timing can be accurately estimated. The interaction of the DSPs and the data exchange based on the module times is predicted by the simulation. Consistency of the timing data used by the time/event simulation and the DFG behavioral simulation and the subsequent autocode generated must be maintained.

At a coarse level of accuracy, the Performance Modeler in Figure 5-7 determines the shape of the speedup curve from two functions: decomposition and contention. The decomposition function represents the extra work due to partitioning an algorithm to run in parallel. This extra work is composed of data copying and recomputations. Example decomposition functions that have been observed include N, $\log(N)$, $\sqrt{N}$, and N^2. The contention function represents delays due to demands for the same resource be it memory, bus, or data. Contention functions that have been observed in practice include N, $\log(N,)\sqrt{N}$, and N^2.

In general, partitioning of the Signal Flow Graph onto the proposed architecture is an NP Complete problem. A variety of heuristics have been studied at CMU including node bin packing, coalescing, node plus arc graph packing, and arc min-cut. The result is a mapping of the task to processors. The first step is to convert the Signal Flow Graph into a Utilization Signal Flow Graph as depicted in Figure 5-9. Here the node and arc weights are normalized to the capacity of a single processor. For example, Figure 5-9 was derived from Figure 5-8 by assuming that a single processor was capable of 100,000 operations per second and a 100,000 data transfer per second.

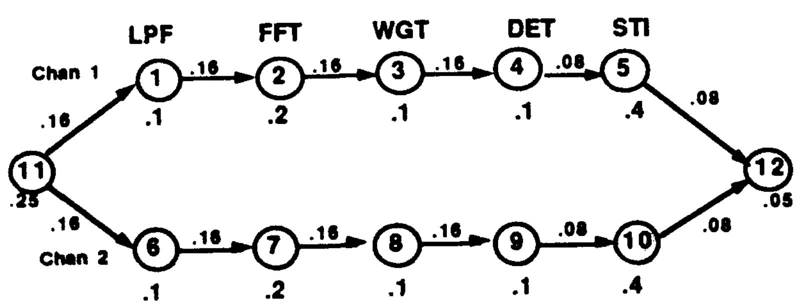


Figure 5-9. Two-channel CR Lofar utilization signal flow graph.

Thus, by analyzing the computational algorithms for a prestored set of functions which will make up the nodes in a Signal Flow Graph, we are able to predict the speedup as a function of the number of processors.

The performance model is parameterized by a number of architectural abstractions including the relative cost of inter- and intra-processor communications, bus bandwidth, processor operations per second, concurrency, and cache hit rates. Initial values for these abstractions will be selected in order for the Signal Flow Graph to meet its performance specification. Subsequently, feedback from the Hardware Synthesis and Software Codesign modules will increase the accuracy of the model.

In summary, there are two basic methods to partition algorithms onto processors. A synthesis method based upon minimizing a cost objective, e.g., least number of DSPs, while meeting latency and cycle time constraints, can be used. Direct synthesis is desirable, but certain constraints and approximations require use of a manual partition, which is optimized via trial and error. With a large number of DSPs, makes trial and error design approaches unwieldy with no assurance that an "optimal" solution is being converged upon. Therefore, RASSP must support a combination of baseline synthesis partitioning algorithms combined with the ability to make incremental changes and verify the partition solution by simulation.

5.3.2 Hardware/Software Codesign

Codesign is often defined as the concurrent development of hardware and software of a system. Hence, codesign encompasses several engineering domains, including architecture design, hardware design, and software design.

A specific approach proposed for the Hardware/Software Codesign (HSC) utilizes ongoing efforts at Carnegie Mellon University and is based on integration of hardware synthesis (such as the MICON/FIDELITY system), an object oriented software development system (to be determined as part of the integration of HSC with other RASSP subsystems), an automated performance/resource software characterizer and a Hardware/Software Codesigner.

The codesign process is shown in Figure 5-10. The hardware design requirements from the Hardware/Software Partitioner (HSP) are part of the input for the hardware synthesis. The software design requirements from HSP are part of the input required by the CASE tools used to develop the software implementation. The Performance Characterizer automatically profiles this software, in the context of the architecture proposed by the synthesis tools. Based on this information, the Codesign Analyzer is making suggestions to the hardware synthesizer to change the proposed architecture.

Furthermore, it is envisioned that feedback from the Codesign Analyzer will also be provided to HSP for further refinement of the hardware/software partitioning and process/processor mapping.

The enabling technologies for the codesigner have been developed as a part of the PIE system. Specifically, two technologies are relevant:

- Automated software performance/resource characterization at the level of task, synchronization/communication and sequential language constructs.

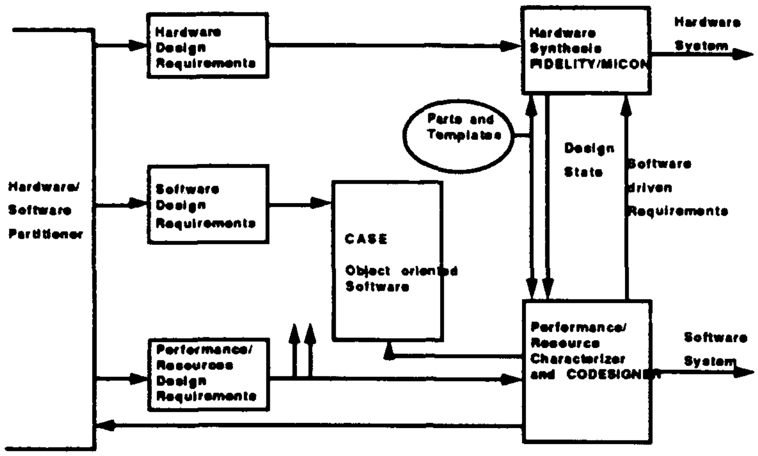


Figure 5-10. Codesign process.

- Development of high-level templates for software architectures. The templates provide both an analytical model as well as a programming template for a number of uniprocessor or distributed/parallel software architectures such as master-slave, pipeline and blackboard. These object-oriented templates have been precharacterized with respect to performance and resource requirement.

Figure 5-11 shows a more detailed view of the codesign process. A library of frequently used objects and signal processing software templates will be provided. Those objects will be precharacterized by the Performance/Resource Characterizer (PRC). The software infrastructure model is assumed to be an open software microkernel technology with real-time, fault tolerance and security (possibly based on current CMU efforts in the Real Time, Fault-Tolerant Mach).

The CASE tools set will be object oriented. The software objects will be characterized by the Programming and Instrumentation Environment (PIE) system on performance and resource metric.

Each such metric will be a pair of required versus attainable performance/resources values, in the hardware implementation proposed by the hardware synthesizer. For example, each object will be characterized by resources such as the ratio of arithmetic/data transfer/control instructions and ratio of float/integer operations. Each object will be characterized by measures such as memory size and I/O bandwidth.

As a result of this process (Figure 5-12), each object's contribution to the software system performance/resource will be translated by the Codesign Analyzer (Figure 5-11) into a set of proposed system changes to be input to the Hardware Synthesis.

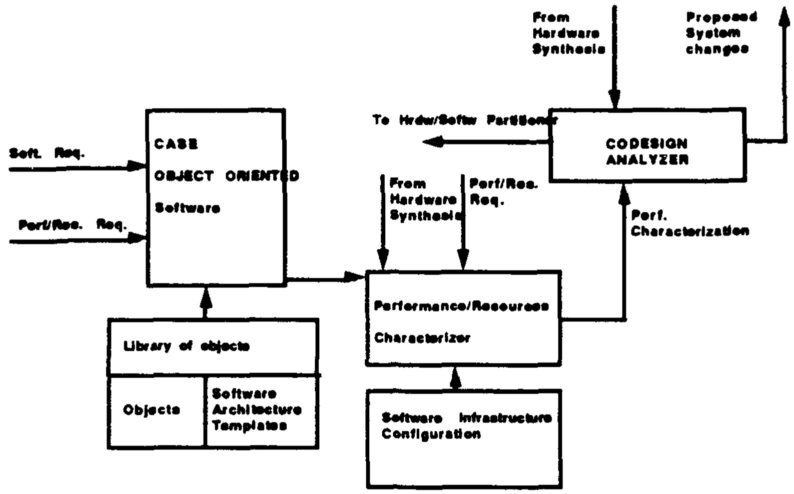


Figure 5-11. Codesign process details.

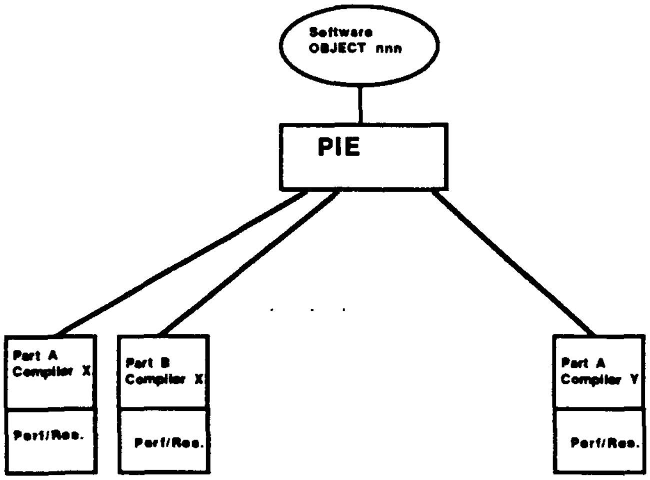


Figure 5-12. Library of objects and their perf./res. characterization.

5.3.3 Architecture Selection

Architecture selection is the development of the basic hardware and software structure of the signal processing system in terms of the number, type, and interconnection of the processing units and their interfaces to related subsystems, and the scheduler/operating system organization. For each processing unit within a parallel signal processor, the functions to be performed, the required performance, and the number and type of interface ports is defined. The required performance is usually expressed in terms of the integer or floating-point operation rate determined from the application algorithms in the algorithm definition phase. The memory requirements and location must also be determined for each processing unit. Initially, the amount of shared and local memory is estimated from the data storage requirements of the application algorithms plus the prospective operating system overhead. Tradeoffs are made within the mapping task on jointly minimizing the communications and the data partitioned throughout the system.

An interconnect structure is chosen such as a bus, ring, star, mesh, switch or a custom combination of these based on the tentative communication patterns of the application algorithms. The required bandwidth and message latency are also stated for each network interface or linkage. The choice of network architecture, link types, or communication protocols may depend on the average, worst case, and variance of the message latency that can be tolerated by the algorithms and their mappings. The network effectiveness is tested using the architecture simulator.

The initial architecture and system sizing is based upon estimates of the required processor and communications throughput from the algorithm and system concept definition and upon estimates of available processing element performance on the application algorithms, as described in Section 5.3.2. As the architecture selection process proceeds, better estimates become available through DFG and architecture simulation and processor benchmarking. The architecture is modified accordingly and re-analyzed. The architecture simulation identifies bottle-necks, and determines processing utilization, load balance, overall processing latency, resultant system throughput, and other deficiencies in the architecture. This process iterates until the specified signal processor system requirements are satisfied.

The signal processor architecture may be constructed from standard architecture templates that are maintained within the design library. Customizations can be specified directly. Specialized expert design advisors could accelerate and help to further automate the architecture specification process by analyzing the requirements and simulation data and suggesting architectures and/or modifications. Such advisors are driven by specialized knowledge bases in the architecture area and have access to existing architecture design libraries. To facilitate rapid development, it would be helpful to have a process management tool which oversees the architecture selection process. The selection tool should invoke the advisor and the simulator in response to feedback between the these tools and the designer. The selection tool should also maintain records on the investigated architectural options.

5.3.4 Architecture and Data Flow Graph Simulation

The architecture and data flow graph simulation models the execution of the data flow graph on the prospective parallel architecture. It produces feedback on the effectiveness of the mapping, scheduling, and architecture combination. Specifically, the simulation provides information on the processor element and communication link utilization, loading and load balance. It identifies bottle-necks in the architecture, and it determines if the system processing and throughput requirements are satisfied. The simulation can be used to analyze the transitions between algorithms and system modes in the signal processor system. Examples of DFG-based tools of this type are Comdisco's SPW, Mentor's DSPStation, and GE's Distributed Application Environment (GEDAE).

The architecture simulation may accept the DFG produced produced by the system concept/algorithm definition task. It also accepts the architecture information, as specified by the architecture selection tool, and the scheduling information prepared by the mapping tool. The simulator produces data which are feed back to the expert advisor, architecture selector, and mapping tool. It also produces timing and control information that may be used in the lower level hardware and software designs.

The architecture simulation lends insight and increases understanding about the mapping and architecture performance. The simulator can produce extensive visualization of the DFG mapping and execution on the candidate parallel architecture. For efficiency it models only the time required to perform the operations by the respective processing elements, not the operations themselves. Similarly, the simulation models the time required to gain access and transfer data across network linkages. It implements a resource utilization and queuing model. With it, network timing flow analysis and timeline simulation can be done. Data flow, network simulators of this type include Bones, ADAS, and SES Workbench.

The simulation provides an opportunity to test the architecture and mapping before the hardware is constructed. It provides a means for quickly optimizing the architecture mapping by rapidly assessing modifications to the DFG, architecture, scheduler and experiments with other parameters. As more accurate timing values become known through benchmarking and modeling, they replace earlier estimates and the architecture mapping is re-verified.

The simulator can be used for further testing and optimization by expanding the delay model to include behavioral models in which the actual operations are performed on actual test data that are moved across the simulated network. The sufficiency of the architecture and correctness of the mapping can be verified before continuing on to more detailed aspects of the design.

5.3.5 CAD Tool Requirements by Task

Four classes of tools are required for the RASSP architecture definition process. These tools include the architecture selector, the partitioner/mapper, the architecture selection/mapping advisor, and the architecture simulator. The requirements for each of these tools are described below.

Architecture Selection: The architecture selection tool should operate with a graphical block diagram editor for specifying and modifying the DFG and architecture files. It should be a process management tool which controls the architecture selection process. The selection tool should invoke the advisor and the simulator in response to feedback between the these tools and the designer. The selection tool should maintain records on the investigated architecture options. It would be desirable if the selection tool and/or the related advisors could utilize the information in the records of previous experiments. The capability of intelligently utilizing simulation results would be desirable, as would access to an automatically updated design data base of node execution times. The selection tool should interface to existing design libraries of architecture templates, and its output should be interpretable by the design advisors, the architecture simulator, and the DFG mapper.

Expert Architecture Selection/Mapping Advisor: The design advisors should accept an understanding of the required goals and recommend appropriate mapping and scheduling strategies. They should be driven by specialized knowledge bases in the architecture area and have access to existing architecture design libraries. The advisors should analyze the requirements and simulation data and suggest architectures and/or modifications. The advisors should analyze or guide the evaluation of alternatives based on existing designs.

Architecture Simulator: The architecture simulator should provide flexible means for extensive visualization of DFG execution mapping on the architecture and performance indices such as queue lengths, resource utilizations, and contentions. The simulator should operate compatibly with other tools. It should accept DFG and architecture descriptions from the system concept/algorithm development tools. The simulator should specifically support multi-processor architectures. The simulator should also facilitate checking for requirements satisfaction from data inherent in the test cases. It should provide a graphical block diagram editor for specifying and modifying the DFG and architecture. The data produced by the simulator should be compatible with the lower level tools such as for static schedules and hardware architecture structure. The simulator should be capable of executing behavior models underlying the nodes. It should be compatible with lower level simulators for mixed-level hierarchical co-simulation.

Partitioning/Mapping: The partitioning/mapping tool should interface to default scheduling discipline libraries. It must possess an expert advisor interface. The scheduling/mapping tools must be compatible with the architecture simulator. It should produce scheduler files that the simulator can accept, and it should accept simulator result data. The mapping tool should produce the developed schedules and scheduler code which can be directly implemented on processors via auto-code generation.

5.3.5.1 Summary of Available Tools

Many vendor tools were surveyed during the first phase of the RASSP program. Data on the tools were collected from vendor demonstrations and product brochures. Time did not permit full evaluations of each product. Table 5-5 summarizes GE's understanding of the current tools which support the signal processor architecture development.

5.3.5.2 Required Developments

Emphasis is placed on areas that will produce the greatest benefit toward RASSP. The following areas are recommended for development under the RASSP program.

Architecture Selection Tool: No such architecture tool is known to exist. However, many of the constituent components such as data bases, framework tools, and version managers exist or have been demonstrated. Therefore, GE recommends development of an architecture selection tool based primarily on integrating existing components.

Table 5-5. Current architecture development tool capabilities.

Requirements	SES Workbench	Foresight Nuthena	Comdisco SPW	Comdisco Bones	GEDAE	CMU's SysArch	Micon Micgen
Partition/Mapping							
- Auto Partition						C	
- Auto Map			L		L		
- 0-0 Prog Sup			L		C		
- Multiprocessor	L	L	L	L	C		
Optimization/Verif.							
- Network Sim	C	C		C			
- Mixed-mode		C	L				
Functional Verif.							
- Hi-Lev Behav		C	C	C			
- Exprt HDL, SDL	L	L	L				
- Flexible Des Sup		C	L				
- Custom						N	
- Library Based						C	
Hierarchical Tool Links		L	L	L			C
Link to Auto Code Gen	L	L	L		L		
Des Advsr/Synth						C/P	P

Legend:
C = Has Capability
P = Planned
L = Limited Capability

Partition/Mapping Tool: A number of partitioning tools are evolving at the University level, including Berkeley's Ptolemny tool and the CMU tools previously described. Extension of these tools and integration into a general design framework is recommended for RASSP. There are, however, no general purpose algorithm mapping systems. However, automatic mapping systems have been created for specific applications such as the many vectorizers, and APPLY and ADAPT from CMU. The technology for a general purpose automatic mapper is not mature. Therefore, GE recommends extending existing mapping systems to meet RASSP requirements.

Architecture Simulation: Many architecture level simulators exist. However none appear to offer all the features required by RASSP. Many do not support multi-processor simulation well. Therefore, this task should select the most appropriate simulation tool and incrementally develop the extended capabilities. Such capabilities may be for instance, compatibility to the other tools, enhanced visualization, and hierarchical mixed-mode co-simulation.

Another required development is the integration of the DFG simulation with the time/event simulation to be able to make use of the DSP timing and data communication models. This process must be complete to include the autocode development, the compilation to the target DSP type, timing evaluation of the particular module and integration of the timing data into the time/event simulation. The process is best facilitated by an approximate timing data library that has been parameterized for the particular family of DSP and the communications models. This library data can be used for the initial timing trade studies. As the tradeoffs are further refined, the autocode path (described in Section 5.5.3) can be invoked to obtain the more refined data. A similar path is needed for the ASIC models. A library of standard ASIC models is first built to facilitate the first level trades. As the design is refined the behavioral functionality will be synthesized from the VHDL description down to the gate level wherein the accurate time data can be obtained. The integration of the data handling of the information is part of the RASSP data base integration task.

Design Advisors: No design advisors are known to exist in this area. However, design advisors have been demonstrated in other applications. Therefore, this task is to select a design advisor system, and create an architecture selection knowledge base. Later, a similar knowledge base for mapping technique would be generated.

5.3.6 Design Language/Information Representation Concepts

The RASSP concept relies on efficient bi-directional information flow between design tools. The current design process requires time-consuming and error-prone translation between tools and domains. VHDL was intended as a standard modeling language within the behavioral, functional, RTL, structural domain. However, RASSP could benefit if languages/formats were to evolve which are capable of spanning other design domains as well.

The available RASSP resources are certainly not sufficient to develop such new languages/formats, and therefore is likely to identify several description languages to support the design system. Efforts would then focus on encapsulating these languages into a common framework.

Thus RASSP will suggest or support extensions in specific areas that may have the greatest impact upon rapid prototyping. Eventually, such extensions may be officially accepted. IEEE Std-1076 VHDL has undergone extensive scrutiny for improvement and extension for its re-standardization due in 1992. However, the slated changes tend to be lower-level and syntactic in nature, without adding much functionality. Most analog extensions are currently being deferred.

The following discussion, describes areas in which the current design languages and the tools that use them could be improved. Below are listed some of the critical deficiencies and potential remedies of IEEE-1076 VHDL relative to the RASSP requirements. Some of the remedies have been previously suggested and are currently under review or have been deferred.

5.3.6.1 DSP Subsystem Design Support

DSP system design requires features which are deficient in current the hardware description languages. The following discusses some of these issues.

Means to Convey Mechanical/Physical Design Data with Models: *The conveyance of* physical and mechanical design data *with electrical behavioral/logic models is* perhaps the weakest area in current design technology. Certainly there are several formats for these aspects, but no language seems capable of conveying all the appropriate information, and they tend not to be tightly integrated into the design tools throughout the process. The information is maintained and processed separately. Therefore, it would be desirable to integrate into the language a standard format for conveying design data from domains other than just the behavioral, functional, RTL, structural domains. For instance, methods are needed for describing the electrical domain concepts of voltages, currents, temperature, and power; the physical domain aspects of packaging, pin-outs, weight, volume, dimensions, mechanical, materials, and manufacturing processes; and the timing domain of min/typ/max propagation delays and constraints, rise/fall time characteristics, and loading.

There are already improvements in the VHDL language capabilities in this area designed into the proposed 1992 revision. PAP-E ReManufacture Application Protocol will provide a methodology linking simulated behavior to these other information areas. Back annotated timing is being standardized in an IEEE sub-committee.

Although some data is adequately conveyed by other standard formats, such as Gerber geometric data, standard formats are needed for other physical data. One approach is to develop a special format for each kind of physical data which is then translated into VHDL by a preprocessor.

<u>Scientific Variable Types:</u> For compatibility with the higher design levels, several scientific variable types would be useful. For instance, complex, floating-point, double precision, user definable structures, and arrays, along with the corresponding operators to operate on these types, such as complex multiply, would be useful. Some of these are now available in some HDLs, and RASSP is but one of many desiring these types. Some of these concepts are under review by the IEEE DASS. If not included in the language definition, these could be addressed by standard libraries and packages.

<u>Scientific Functions:</u> For use at the higher design levels, common access to several abstract function types would ease development. For instance, useful functions would be transcendental routines, exponentials, powers, roots, Z-transforms, Laplace Domain Analysis, FFT, and the library routines from BLAS, EisPack, LinPack, IMSL, and LAPACK. Additional routines would be algorithms for searching, sorting, hashing, tree traversal, search, and construction, graph traversal, spanning tree computations, polynomial arithmetic and Galois operators. Carefully written parameterized functions would be a welcome addition to the standard functions VHDL provides.

One difficultly is not in writing these functions/algorithms in VHDL, but in providing a suitable medium in which the user can browse, access, and add VHDL source, documentation (including graphics), usage notes, and suggestions on closely related objects. A source code library manager (SCLM) could aid in doing these things. Another possible solution is to produce a pre-processor that would allow the use of scientific data types and functions as though they are built into VHDL, without the tedious libraries references.

<u>Procedural Programming Mode:</u> Since VHDL is intended for describing hardware behavior, the descriptive paradigm is a mapping or transform between the inputs and outputs. This assumption causes many artifacts of the language definition to make modeling the higher level abstractions less convenient and less natural. For instance, within a VHDL behavioral block is an inherent unconditional loop. However, model for

the algorithm concept level typically are sequential programs with a global main that runs once. Methods to also support this descriptive level would be desirable.

One approach is to work within the CFI Simulation Backplane Working Group to make sure that their standard covers such linking between pre- and post-timesweep global computation. Support must include separate compilation of functions and procedures, with makefiles and etc. A VHDL pre-processor or even a whole family of them could enforce rules such as a single "main" function, allow the declaration of ".h" files, and compilation commands to generates proper VHDL packages and declarations.

Object Oriented Extensions: Object orientated construction is a modern language technology that is often used for abstract higher level modeling. However, it may be useful throughout all levels of the design process. For instance, it may be useful to introduce class hierarchy and inheritance of methods for entities, ports, signals, generics, and etc.. A pre-processor could be used to translate these into standard VHDL. This would be similar technology to that of Obj-C pre-processors.

5.3.6.2 Virtual Prototyping Support

Although all VHDL simulators are interactive, the user can affect the simulation only at breakpoints. This is not sufficient for virtual prototyping applications. Means for coupling simulations with graphical interfaces such as X-windows for manipulating and experiencing simulated systems with sliders, buttons, viewers, and meters, would be desirable. Such capability should be a tool or framework feature, so no language change should be necessary.

A fairly primitive model of support for virtual prototyping is possible which will not require major changes in any VHDL toolset. However, a foreign language interface is recommended. This will not involve any changes to VHDL syntax, but will involve changes in VHDL environment tools during the model generation and build phases.

First, we need to provide a truly interactive method of VHDL simulation. In general simulations, the Application generates events that are processed either in the application process itself or sent over to the VHDL process for event handling. This configuration can be achieved by running both the VHDL model and the application code as child processes of a parent process. A better solution is to use IPC or (even RPC). This can be achieved through a foreign language interface to VHDL. This is an ability to call functions written in other languages from within the VHDL code. The syntax of the language would not change in any way. We could declare a VHDL function interface to the foreign function, but specify the function body as a compiled object that would be linked during the model generation/build process. Similar facilities are available from the MCC VHDL simulator for linking circuit level behaviors with gate level component instantiations in VHDL.

5.3.6.3 Hardware/Software Co-design Support

Hardware-software co-design can significantly reduce RASSP development time by ensuring the early integration of the software with the hardware (see Section 5.2.3). There are limits to the extent to which VHDL should be extended to address software.

A hardware descriptive language such as VHDL cannot be expected to convey software description. Therefore, software descriptive language (SDL), with associated development tools are needed. Standardized Software Description Languages (SDL) do not exist today; a SDL is most likely to look like pseudo-code on Process Description Language (PDL), from which a HDL can be synthesized. This allows the user to generate code for a number of HDL languages (C, ADA, etc.) from the same description. Such tools must accept and operate with target software code, such as output from the auto-code generators.

RASSP needs simulation support for software development, such as meta-assemblers, meta-compilers, symbolic debugger interfaces, profilers, and simulated memory structures. The DARPA DSSA and ProtoTech progress should be reviewed before selecting an approach to compiler-compiler tools for this area.

Some of the required technology is available, such as Zycad's N.dot meta-compilers, and meta-assemblers. The RASSP requirements could be realized through selection, integration, and extension of existing tools. This is an area requiring further study.

5.3.6.4 Mixed Analog/Digital Design Support

A method of modeling mixed analog and digital systems is very important for RASSP. Intermetrics will soon be completing the definition of a microwave hardware descriptive language (MHDL). It is based upon VHDL, but contains analog/microwave modeling extensions. Analog extensions to VHDL were deferred until MHDL is defined and specified.

Since MHDL is intended for spatially distributed-parameter microwave circuits, it may not be appropriate for another lumped-parameter system. Since no modification to VHDL is expected in the foreseeable future, RASSP is likely to require an interim solution.

Some capability is available commercially. Analogy Inc. has demonstrated the apparently viable mixed analog/digital technology in its MAST tool sets. Analogy is promoting an AHDL that is consistent with VHDL from its MAST toolset. A RASSP effort should devise a means for evaluating the quality of mixed analog-digital simulation systems. An attempt should then be made for evaluating the capabilities of these tools, and possibly select and integrate them into the RASSP toolset.

5.3.6.5 VHDL Modeling Support

The RASSP effort would significantly benefit from the greater availability of more VHDL models. This could eliminate much of the modeling time which is often consumed for developing new models of every component for each design. Certainly it is not cost-effective even for every company to create its own library of VHDL models.

Modeling is being addressed by a variety of sources. The IEEE DASS has a section on modeling and it is working with the EIA to adopt standards in style. The similar VITAL effort is also looking at ways to standardize modeling styles. RASSP could promote or initiate the formation of a public domain resource for standard VHDL models of standard components at various modeling abstractions. The library might make free model available, and it might provide an index of licensable models. The models could have various levels verification, such as, verified by authority or committee, verified by user(s), and unverified.

5.4 Hardware Design

The digital hardware design is based upon the component descriptions developed under the architecture development phase as described in Section 5.2.2. In the digital hardware design phase, a hardware realization is selected for each of the digital components. Where applicable, commercial off-the-self (COTS) modules are selected. In other cases, boards may be configured as a custom combination of COTS modules, and/or fully custom boards or chips may be required. The COTS modules, configurations, and/or custom designs are specified. The process is aided by knowledge based advisors which have access to data bases of available application specific modules. COTS modules must be specified for procurement, while custom modules must be designed for manufacture. In either case, all modules must be modeled for joint hardware/software simulation, system development, and requirements verification. For rapid design, custom chips and modules must exploit automatic synthesis and layout technology. The design must be analyzed for thermal, mechanical, reliability, and maintainability requirements. The design must incorporate built-in test and design for test techniques. The result of digital hardware design is the simulatable model, procurement orders, manufacturing data, and unit and system tests.

5.4.1 Digital Hardware Design

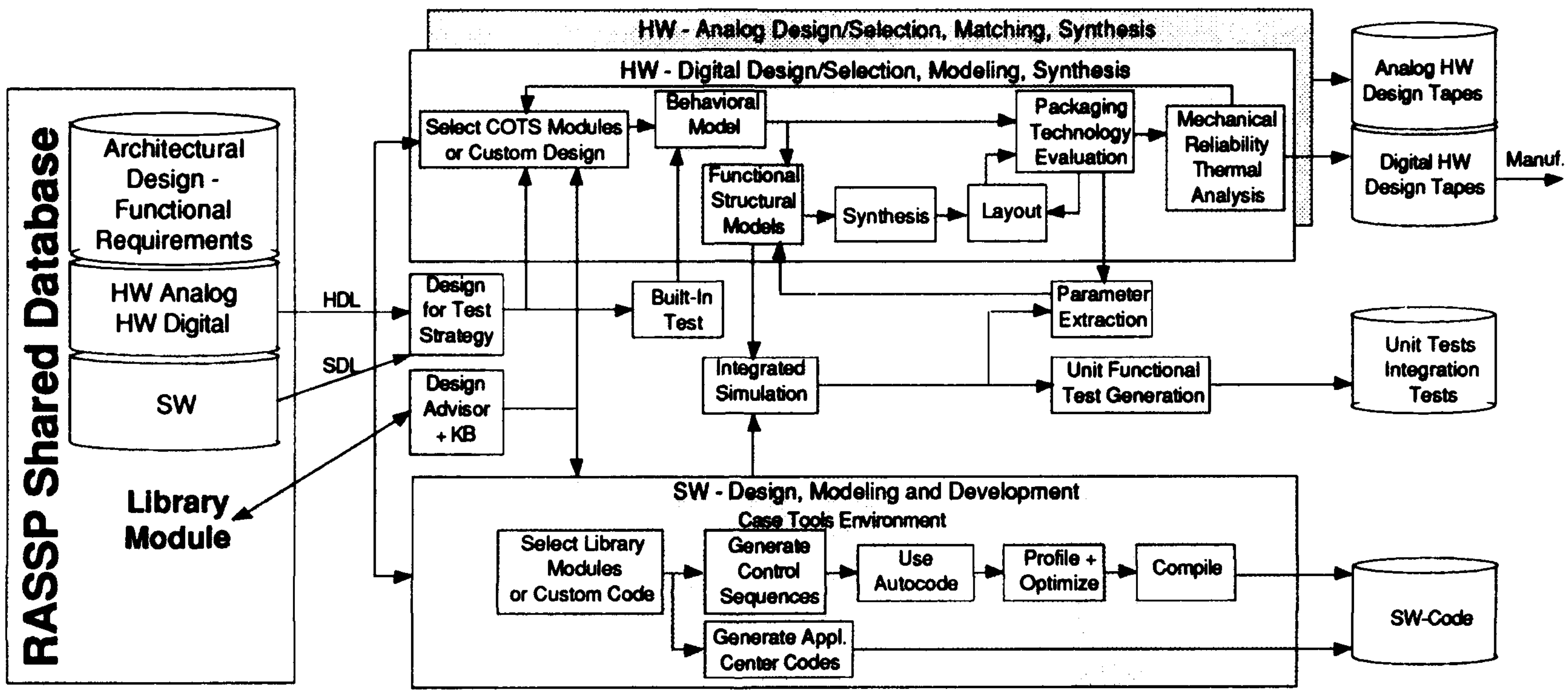
5.4.1.1 Required Tasks/Functions

The digital hardware design phase is divided into eight tasks, as shown in Figure 5-13: test generation, COTS or custom module selection, behavioral modeling, design for test and built-in test injection, functional/structural modeling, integrated simulation, deign synthesis, and mechanical-reliability-thermal analysis. Each task is described below.

Test Generation: The test generation process decomposes the component specifications into a set of tests which concisely check for the component's satisfaction of these specifications.

Test patterns are synthesized which pair expected result patterns with the stimulus test patterns. For rapid generation, most of the stimulus patterns can be obtained from the other models or modeling tools. Similarly, many of the expected response patterns can be obtained from the higher level models, such as from the algorithm or architecture model simulations. This helps maintain consistency across the modeling levels. The test data are used to drive the related analysis and simulation tools.

The test should include checks for all types of requirements data, not only electrical/behavioral/ logic, but also physical/mechanical. The consistency and



Concurrent Joint HW/SW Development
- Staged Joint Analysis, Simulation, and Construction provides Early Integration, and Quick Design Feedback for Rapid Prototyping

Integrated Simulation
- Hierarchal Mixed-Level Simulation
- Early System/SubSystem Integration
- Soft test Bench/Lab
- Virtual Prototyping Environment

HW Design, Modeling and Development
- Select COTS Modules or Custom Design (Trade-off Data and Tool)
- Aids for Developing VHDL Models that drive automated synthesis of Custom Blocks

SW Design and Development
- Select COTS Modules or Custom Code
- Early and continuous Integration with System Hardware Modules
- Use Auto-Code Generation

JSP 57

Figure 5-13. RASSP component HW/SW development process.

completeness of the requirements should be checked throughout the test development process.

COTS or Custom Module Selection: The module selection process is driven by component requirements accepted from the architecture design phase. The signal processor components may be processing elements, memory modules, buses, interconnection networks, controllers, and I/O units. The requirements include such parameters as data transfer rates, buffer sizes, memory sizes, number of ports, instruction and floating-point execution rates, physical size, and software/electrical compatibility. To eliminate component development time and cost, COTS parts are selected where applicable. This could be accelerated with the aid of design advisors which access libraries of COTS modules. COTS parts may include general purpose or special function chips, modules, boards, and sub-systems boxes. When COTS parts are needed, procurement orders must be generated and passed to the procurement process.

Despite rapid increases in COTS performance, many military requirements remain far ahead of current COTS technologies. Such cases require a custom or an application specific design. The selection process must identify such cases, and if possible recommend custom configurations of COTS parts. For example, an application specific board composed of COTS processing chips may be needed to satisfy military processing density or packaging requirements. When custom hardware is needed, the requirements must be passed on to the custom hardware design process.

In either the custom or COTS case, a behavior model should be obtained for integrated simulation and testing. In the custom hardware case, a model must be generated, while in the COTS case, standard models may be available from libraries.

Behavioral Modeling: Behavioral modeling of each system module is required for early integrated simulation and testing, software co-development, rapid design feedback, and virtual prototyping. In the case of custom hardware, behavioral modeling is one of the first steps in the hardware design process. Behavioral model generation can be accelerated by providing means for extracting modeling information from the algorithm and architectural description levels. It can be further accelerated by adopting standard modeling practices and interfaces which ease the design of models that must operate cohesively.

Design for Test & Built-In Test Injection: It is very important that design for test concepts are included at a very early stage in any custom design. The design for test strategy selection is based upon the hardware and software requirements. The strategy affects the module selection and test generation processes. Consequently, this selection should be one of the first activities.

The built-in test capabilities should be injected jointly into hardware and software. In the digital hardware, the built-in test should be injected into the behavioral level design and models. This ensures that the design for test strategy is consistent with the design, and maintains its consistency as the design progresses. It also helps ensure completeness in the testing. Automatic injection of built-in test capability currently exists at the lower logic and circuit design levels. To save time, standard built-in test injection could be automated at this higher level.

Functional/Structural Modeling: After the behavior level design of the custom components is completed, the functional and structural level is designed and modeled. These models are integrated with the rest of the system in the form of mixed-level simulations.

For efficiency and correctness, much modeling information is extracted from the behavioral level models. Additional time could be saved by automatic synthesis of the structural models from the behavioral models. Such synthesis capability is currently mature only at the logic and gate levels. New tools are extending the capabilities to higher levels, such as RTL. Such capability could be driven by comprehensive application specific design libraries.

The structural models of custom chips are passed to gate level synthesizers in the design synthesis phase. Models composed of custom configurations of COTS parts pass the configuration data onto automatic layout and routing tools, while the COTS specifications are passed to procurement.

Integrated Simulation: Throughout the design process, simulations are performed of the design elements within the signal processor system at large. These simulations must accommodate both the evolving hardware descriptions and the evolving software. The integrated simulation is the primary support for hardware/software co-design. It ensures not only the early integration of software and hardware, but also the early integration of all hardware component designs.

Execution efficiency is maintained through mixed-level simulation. In mixed-level simulation, the module of interest is simulated at the lowest available detail level, while other system modules are simulated behaviorally. The integrated simulation requires a common (or compatible) design descriptive language(s).

The integrated simulation tests for design effectiveness and for requirements satisfaction. It produces early feedback on trouble spots in the designs, such as incompatibility between modules. This allows rapid correction and design modification, and it precludes time consuming lower level re-designs and re-builds.

Design Synthesis: In the design synthesis phase, structural or RTL level models are passed to logic synthesizing tools. These tools synthesize space and time efficient ASIC circuitry in a variety of technologies. Synthesized designs, which are correct-by-design, reduce the total debugging time. The resulting logic level and macro-cell design is passed directly to automatic board and chip layout tools. These tools produce data in formats which drive the chip and board manufacturing process.

Mechanical-Reliability-Thermal Analysis: The design is analyzed in several ways as the physical design information becomes available and especially after layout. Thermal analysis is performed to check for hot spots and provide feedback on modifying the design to avoid them. Mechanical vibration analysis is performed to detect destructive resonances and stress points. Reliability analysis provides feedback on the expected MTBF of the system based on the rated MTBF of the components and the configuration. Other analysis, such as packaging technology evaluation, should also be performed. The feedback is used to modify the design and avoid problems at an early stage before construction to avoid time consuming re-builds.

5.4.1.2 CAD Tool Requirements Per Task

Nine classes of tools are required for the RASSP digital hardware design process. These tools include the test generator, design advisor, COTS module selection aid with procurement interface, design for test and built-in test injection aids, integrated simulator, automated synthesizer, automated layout and router with manufacturing interface, framework, and mechanical-reliability-thermal analyzers. The requirements for each of these tools are described below.

Test Generator: The test generation tool must accept the component specifications and aid in decomposing them into a set of tests which concisely check for each component's satisfaction of these specifications. The test generation tool must provide capability to synthesize test patterns.

The test generator must provide means to pair expected results with the tests. It should accept compatible data from the other modeling tools such as the architecture level modeling tools. The test generator should provide means to incorporate some of the higher level algorithm and architecture tests into the component and board level tests. The test generator must produce output test format which is compatible to the related analysis and simulation tools.

The test generator should make use of standard formats for conveying all types of requirements data, not only electrical/behavioral/ logic, but also physical/mechanical. The test generator should check for requirements consistency and completeness in the generated tests.

Design Advisor: The design advisors should accept an HDL descriptions of the design requirements and recommend appropriate COTS solutions and design-for-test strategies. They should be driven by specialized knowledge bases for test strategies and COTS libraries.

COTS Module Selection Aid with Procurement Interface: The module selection tool should work in concert with the design advisor to access the COTS libraries for appropriate components. Factors considered in selecting modules are cost, performance, size, weight, power, and support. The selection aid should guide the designer through the custom, semi-custom, or COTS solutions. When a COTS module is selected, an interface to procurement should supply the appropriate order information such as supplier, model number, and lead times.

Design for Test and Built-in Test Injection Aids: The design-for-test strategies tool should guide the designer through the process of selecting an appropriate design-for-test strategy for the given architecture, and help in implementing it by providing appropriate test functions from design the library.

In conjunction with the design-for-test strategy, the built-in test injection aid should aid the designer in selecting appropriate built-in test circuitry for the sub-system, board, module, and/or chip level. It should then provide means to automatically insert such built-in test functions into the design. The built-in test injection tool should also generate the test vectors for exercise the test functions.

Integrated Simulator: The integrated simulator must accommodate hierarchical mixed-mode simulations. It must jointly accommodate both the evolving hardware descriptions and the evolving software. The integrated simulation must use a common (or compatible) design descriptive language(s) for all models.

The integrated simulation should accept the generated test formats. It should use these tests to check for requirements satisfaction. It should also check for the consistency of some of the physical requirements that are contained in the test data and in the models.

It would be desirable for the integrated simulator to operate compatibly with the other tools. It should be compatible with the higher level simulators such as the algorithm, architecture, and analog simulators for hierarchical mixed-level co-simulation.

Automated Synthesizer: The automated synthesizer should accept structural, and desirably functional, descriptions of modules and synthesize time and space efficient logic. The synthesizer should produce optimized logic for a variety of technologies. It would be desirable for the synthesizer to produce board and module logic in addition to ASIC logic. It would be desirable for the synthesizer to have the capability to compose designs of circuit modules higher in level than logic gates, such as macro-cells, COTS chips, and chip modules. Synthesized designs, which are correct-by-design, reduce the total debugging time. The ability to automatically synthesize RTL descriptions from behavioral descriptions would also be greatly desirable, since it would vastly accelerate application specific signal processor design.

Automated Layout and Router with Manufacturing Interface: The layout and router tools should automatically place and route ASIC, chip module, and board logic efficiently and quickly. They should require little supervision. The should employ programmable design rule checkers. The output of these tools should be compatible with standard board, module, and chip fabrication formats.

Framework: The framework tool should integrate the operation of all tools. It should aid the designer in invoking and transferring information between tools. It should provide a consistent user interface across tools.

Mechanical-Reliability-Thermal Analyzers: The mechanical, reliability, and thermal analyzers should operate at the module, board, and sub-system box level. The analyzers should operate at several levels within the design process, from the early conceptual, architectural, structural levels, down to the mechanical design level. At the upper levels, they should operate on tentative component use, density, and construction data for feasibility analysis. The analysis tools should be compatible with the other design tools, especially in accepting the design information in the form of the common design description languages. They should also be invoked by the framework, and communicate results back to the other tools.

5.4.1.3 Summary of Available Tools

Many vendor tools were surveyed during the first phase of the RASSP program. Data on the tools were collected from vendor demonstrations and product brochures. Time did not permit full evaluations of each product. Table 5-6 summarizes GE's understanding of the current tools which support the signal processor digital hardware development.

Table 5-6. Current digital hardware development tool capabilities.

Hardware	Analogy	Mentor	Cadence	Zycad Protocol	Dazix	Logic Modeling	Micon CMU/ Omniview
Functional Simulation	C	C	C	C	C	C	L
Mixed Mode	L	L	L	L	L	L	L
Link to Software							
- Electrical Description	L	C	C				C
- Mechanical Description		C			C		L
- DFT Support		C	L	C	C		C
Link to Manufacturing		C	L	L	L		
Design Advisor Support							C

Legend:
C = Has Capability
P = Planned
L = Limited Capability

5.4.1.4 Required Developments

Most of the tools required for RASSP digital hardware design are currently available. However, a few of the critical features required in these tools do not exist. Some of these capabilities are emerging naturally in the commercial marketplace. The following areas are recommended for development under RASSP program, since funding them will provide the greatest reduction in DSP hardware development time.

Test Generator: There are many test generators at the logic level, but test generation at the behavioral level appears to still be ad hoc. There are several requirements analysis tools at the higher levels, but none for the digital hardware design levels. There are currently no standards for embedding all requirements data (including physical and mechanical) into automatic tests.

Most of the components for a test generator as described in Section 5.3.1 exist and have been demonstrated, such as requirements test pattern generators for logic, and requirements consistency checking. Integrating them into a common tool and extending the critical aspects requires development. For instance, a means must be reached for encoding the physical and mechanical information into the test. A task which integrates these components and develops the necessary aspects would have the highest payoff for RASSP because so may other aspects of the digital design process in RASSP depend on this automatic requirements testing capability.

<u>Design Advisor:</u> No design advisors are known to exist for COTS module selection or digital hardware development. Though, advisors have been demonstrated in other applications. Such an advisor would help to automate the selection process and increase the use of more COTS components in designs. Since this would significantly reduce hardware development time, a task is recommended to select a design advisor system, and create an COTS component library knowledge base specialized for DSP systems. Later, a similar knowledge base for DSP hardware development and design-for-test techniques would be generated.

<u>COTS Module Selection Aid with Procurement Interface:</u> A COTS module selection aid and procurement interface would reduce the time spent looking for specific COTS modules and for acquiring required COTS parts. Whether used manually or in conjunction with an advisor, the time saved would cut the RASSP hardware realization time. Such a tool is essentially a data base retrieval program. The underlying data base tools are currently available. Since RASSP design time can be reduced by simply establishing such a data base and front-end that is compatible with the design advisor, a task is recommended to set-up such a tool.

<u>Built-in Test Injection Aids:</u> Built-in test injectors exist at the logic level. The design of complicated DSP systems would be further advanced by extending automatic test injection up into the RTL or behavioral levels. Therefore, GE proposes an effort to select an existing test injection tool that is compatible with the related RASSP tools, and to extend its capability at the functional level.

<u>Integrated Simulator:</u> The simulator is the core of the hardware development process. It is most critical especially for the rapid design of complicated systems. For a given level of effort spent on improving any tool, extensions to the simulator have perhaps the most potential to greatly speed-up signal processor prototyping.

Several good behavioral and logic VHDL simulators exist. Among them, they have many of the required capabilities for RASSP. Consequently, only minor extensions are needed in this area. Therefore, the following extensions are recommended that will most greatly accelerate the design and prototyping process.

The existing simulation facilities will be evaluated to select one that is most compatible with the RASSP requirements. The simulator is then to be integrated into the RASSP tool-set. Some of the extensions needed are: the compatibility to accept and automatically utilize the tests from the test generator, the development of mixed-mode compatibility with the analog and higher level algorithm and architecture simulators, the usage of description languages for all levels, and the development of better support for joint software development. In particular, a capability for the simulation to accept software in a software description language should be developed along with the tools to process it, such as meta-compilers, and assemblers. The hosting of the simulator on a hardware accelerator would further quicken prototype realization.

<u>Automated Synthesizer:</u> Synthesized designs, which are correct-by-design, reduce design and debugging steps and dramatically cut the time consumed in developing complicated systems. Extending this capability to higher levels would significantly cut more design steps out of the process. Synthesis technology at the logic level is

substantially mature. Incremental development of these capabilities based on existing tools will significantly speed-up RASSP design. Therefore, GE recommends an effort in which available synthesizer technology is selected, which most closely matches the requirements for RASSP, to serve as a basis for extending the capability into the functional design level. This may entail development of higher level building-block libraries.

Automated Layout and Router with Manufacturing Interface: Excellent layout and router tools are commercially available. They do have limitations that often require some manual over-sight. However, extensive effort has been, and continues to be, devoted to improving these tools in the commercial sector. Therefore, it is recommended that little be expended on further developing these tools other than in selecting compatible tools for the RASSP tool-set.

Framework: The framework is important to RASSP. Good framework environments and tools are commercially available. Therefore, GE recommends a task in which an appropriate framework is selected from available products. The RASSP tool-set should then be integrated within this framework.

Mechanical-Reliability-Thermal Analyzers: Excellent analyzers for system reliability, and thermal and mechanical properties are commercially available. However, the existing analyzers tend to operate on completed design data, while prototype design time could be reduced by applying such analysis earlier in the design cycle. This would reduce the time consumed investigating impractical design alternatives. Therefore, a task is recommended in which commercial analyzers are selected and integrated into the RASSP environment. The existing analyzers should be used as a basis to produce extensions which provide design guidance on potential design alternatives before the design is complete.

5.4.2 Analog Hardware Design

5.4.2.1 Required Tasks

A key portion of signal processing systems consist of low noise analog signal processing circuitry. Design decisions made by the analog engineers have a direct effect on the performance of the system. Over the years the number of qualified analog engineers has diminished, and engineering teams are becoming more reliant on tools that can automate the design process, assist the design engineer in performing his tasks, and provide accurate simulations of designs prior to the fabrication of prototype circuits. For this reason, in addition to providing more efficiency in the design process, more sophisticated analog design tools are required to support the process.

The process starts with the establishment of requirements for the analog subsystem (performed in the systems design processes). The design process involves generation of trial circuit designs (based on informed judgments of the design engineer—possibly assisted with knowledge based tools in the future) for experimentation, and testing of the circuits relative to the required functionality and operating characteristics. Design of the circuits involves interconnection of standard parts, or higher level functions (which are then implemented with combinations of standard parts and other functions).

The test of the circuits is with simulation tools in current or future processes. The desired level of verification is to be equivalent to or exceed the degree of verification achievable with breadboarding the circuits. In fact the process of circuit testing should resemble breadboarding (design of soft test fixture, use of power sources and signal generators, and use of observation tools such as scopes, frequency domain analyzers, etc.).

Provisions in the design process for use of knowledge based design advisors is required, both to improve the efficiency in the process, and also in order to leverage the experience of the design experts. The knowledge base, in addition to representing standard rules, needs to be easily updateable, in order to capture expertise local to particular design disciplines or design teams.

The analog designer also needs to analyze the characteristics of the design, after establishment of physical parameters such as placement of components, selection of board type, line widths, spacings, etc. The required capability is to be able to update or "backannotate" the simulation models with the electrical characteristics associated with the physical layout.

Some components associated with analog designs are not "off the shelf" parts, but are custom designed to a set of functional, and electrical characteristics defined in a procurement specification. The definition of the specialized function, the associated models, and the development of the procurement specification are also common tasks, which need to be supported by the CAD system.

Testing of the developed circuits, and correlation of the results to the simulation results is required for validation of the CAD based approach, as well as the models. Approaches for combined simulation/testbench testing need to be developed.

5.4.2.2 CAD Tool Requirements for Analog Design

The required capabilities of the CAD system for effective support of the Analog design process are as follows:

Powerful Language/Design Representation for Implementing Top Level Designs: The language needs to support hierarchical representation of the design (variety of abstraction levels). In addition the language needs to express other phenomena outside the realm of the digital design languages: drift of circuits with temperature, intermittent noise sources, and continuously varying circuits such as phase locked loops. Design generation modes of graphical entry and /or language entry need to be supported, producing a common representation of the design. The design language for analog support needs a degree of commonality/linkage to digital designs.

Powerful Simulation Capability: Simulation needs to address mixed analog/digital model, needs to support verification in multiple domains (t, s, z), need to support high resolution (fine time step) modes and coarse modes of operation (achieve reasonable runtime).

Design Library Support for Multiple Hierarchical Levels: Robust library of standard elements, and higher level design elements needs to be provided. Capability of design organizations to make additions to the library also needs to be supported. The library approach needs to support large libraries of existing SPICE models.

Design Advisor/Synthesis Tools: Methods for use of knowledge based tools to assist the design process need to be developed. Examples of information include application note information associated with specific parts, specialized test approaches, etc. Design expertise specific to particular design disciplines or organizations needs also to be captured in a form enabling reuse and understanding of design rationale by less experienced analog engineers.

Test/Validation Capability for Simulation Results: Soft Test Bench capability is required for checkout of circuits in an efficient manner, similar in concept to breadboarding. Approaches for validation of results via correlation to real circuits needs to be supported (including test of combinations of soft circuits, and physical elements).

Support for Electronic Specification Generation: For support of custom analog circuit procurement, where the item is specified by information in a procurement specification (ex: an RF mixer, or a SAW filter), capability to generate the specification automatically based on information in the simulation model needs to be supported.

5.4.2.3 Current Tool Capabilities

Many of the required capabilities are available in tools offered today such as the Saber/ MAST system provide by Analogy, as well other vendors.

A summary of specific Analogy supported capabilities, which is considered representative of the leading state of the industry is as follows:

- Filter design support
- Mixed analog/digital simulation
- Graphical design entry/edit capability
- Standard circuit libraries
- Input waveform specification capability
- Analysis tools (in addition to time domain simulation) such as pole zero plotting, bode plotting, etc.
- Specification Generation
- Custom Component Development Support
- Worst Case Statistical Analysis
- Test Support - Soft Bench
- Hardware Test Support - Special Test Equipment

5.4.2.4 Required Development Areas

The primary areas of need for further development to effectively address the RASSP requirements are:

Expansion of Language Capabilities: VHDL extensions to address analog modeling need to be specified and support tools developed, or an existing language for analog design needs to be adopted as the standard, and appropriate linkages to VHDL established.

RF Capability Extensions: To enable more comprehensive simulations in the 100 Mhz to 400 Mhz range, expanded capabilities in RF device libraries, and RF templates needs to be developed.

Synthesis/Design Advisor Technologies: Approaches to capture and store designer expertise in knowledge databases for reuse needs to be investigated and implemented on a prototype basis to prove the feasibility of the concept. Also design synthesis tool development needs to be investigated, also making use of captured designer expertise, or identified best practices.

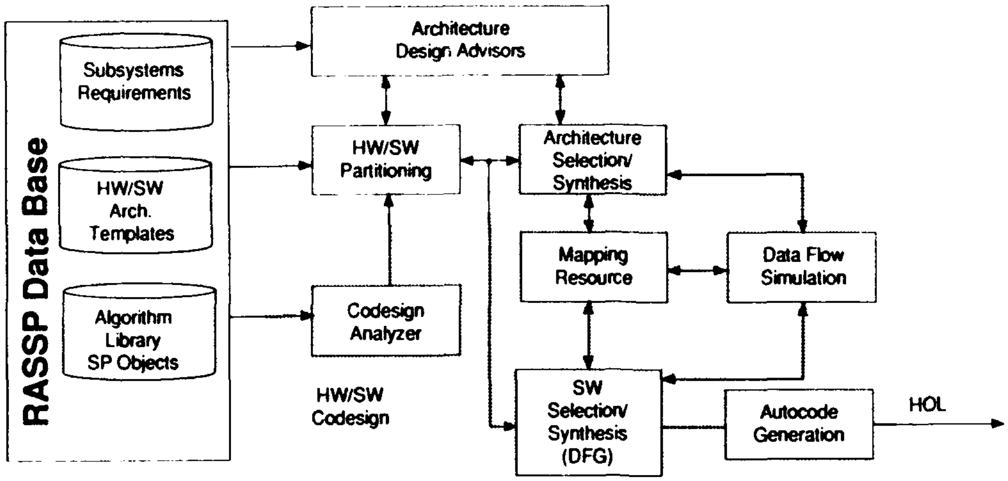
5.5 Software Development

The RASSP software development flow is shown in Figure 5-14 and is very similar, by design, to the architecture development flow. This provides greater support for concurrent hardware/software codesign, which was previously described in Section 5.3.2. This section describes the RASSP software development requirements, how the Data Flow Graph (DFG) tools will be used to develop application code, autocode capabilities needed for RASSP, and the CASE tool support that is required (Sections 5.4.1 - 5.4.2, respectively).

5.5.1 Software Development Requirements

The signal processing software is designed to be part of an overall system. In order to develop the requirements for the CASE (Computer Aided Software Engineering) tools and show how they support the RASSP development of code and its life cycle maintenance the software must be divided into various application domains as explained below. There are three areas that must be defined to show how RASSP is facilitated by these tools. The first is the Application Domain, the second is the life cycle steps and third is the development methodology to be used.

Application Domain: Software associated with signal processing is of an embedded nature and therefore operates under real time constraints. Since, at least in part, the signal processors support high throughput and data flow rates, there is a need for parallel processing. The RASSP system concept and architecture definition incorporates a hardware and software definition phase wherein the software is generated as part of this process. There is a need to support this process with CASE tools.



RASSP software development environment

- HW/SW codesign support
- Data flow graph-driven autocode generation
- Integrated CASE support

Figure 5-14. Software development process.

In addition, signal processors often perform functions associated with Communications, Command, Control and Intelligence (C3I), which involve human interface. This software associated with this is specified by methodologies that follow more conventional CASE design methods.

Life Cycle: In general the steps needed to develop and support software flow include the following: 1) Analysis, 2) Design 3) Coding, 4) Maintenance, and 5) Testing. In addition, the following support services must be available: 1) An environment or framework, 2) Configuration management, 3) Documentation, and 4) Project management.

Methodology: Usually for software development there are three important views of software systems - object oriented (data-oriented), Process-oriented (functional or structured), and behavior-oriented (temporal, state-oriented or dynamic). Each of these views takes a different perspective depending on the software being developed. Conventional CASE tools have been developed to support one of the methodologies that allows the problem solution to be most clearly stated. For example the process of accepting characters from an input device and parsing the input strings is best described in a state-oriented manner. Therefore, a tool that describes and supports diagrams associated with state diagrams is most useful for the purpose of visualization and generation of code to implement this process.

The requirements for software tools are first outlined with respect to the signal processing problem.

The general design methodology that is used for evolving system requirements, developing signal processing algorithms and ultimately to code for the DSP and data processors is based on a block diagram synthesis procedure (see Figure 5-15). The SP tools, in effect, have been specialized for signal processing by using the block diagram or the Data Flow Graph (DFG) as the development paradigm.

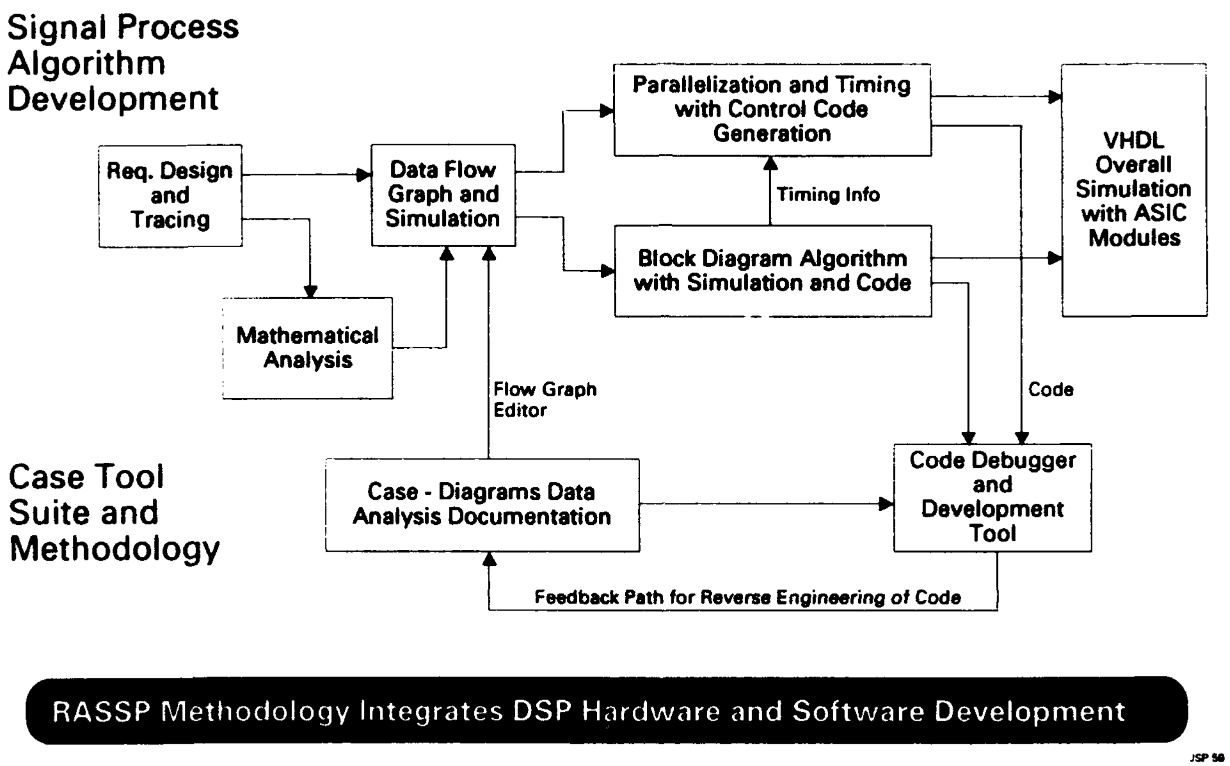


Figure 5-15. Signal processing design with code development.

The CASE tools normally associated with 1) The Data Flow Diagram or (DeMarco/ Yourdon) 2) Data Structure (Jackson) have been replaced by the DFG. The 1) Control Flow (Hartley/Pirbai) and 2) The State Transitions which are usually needed for control software design are also incorporated into the the DFG and the time/event modules is a part of the Block Diagram and time/event editors/ simulator.

Signal processing software is automatically generated as a result of having modules or blocks that represent definable parts of the signal processing algorithms. After partitioning to processing elements, software for the DSP functions is automatically generated. In effect the DFG is supported by compiler technology that generates code for the appropriate DSP machine or machines. The issues such as 1) partitioning to the individual DSP and 2) the communication and control between DSPs are addressed by the DFG generators. The other issue is the integration of the DSP code with the other parts of the overall system. The processors that are used to implement the C3I function make use of data and control processing architectures, for which a conventional CASE methodology is followed for the software design. Data Flow Graph Software development environments and associate autocode capabilities are described in the following sections.

5.5.2 Data Flow Graph Application Development Environment

Figure 5-16 shows an example of Application Development Environment which could support RASSP. Existing capabilities available today include the Interactive Workstation Environment and the Autocode Generation System. The Workstation Environment allows a user to create, modify, parameterize, control, and monitor hierarchical data flow graphs mapped to a heterogeneous set of processors. This is done using primitive functions that tend to be coarse grain (handling on the order of 1000 data samples at a time) in order to achieve efficiency. In contrast, the current Autocode Generation System allows a user, using the same flow graph interface, to generate efficient code to implement very fine grain data flow graphs.

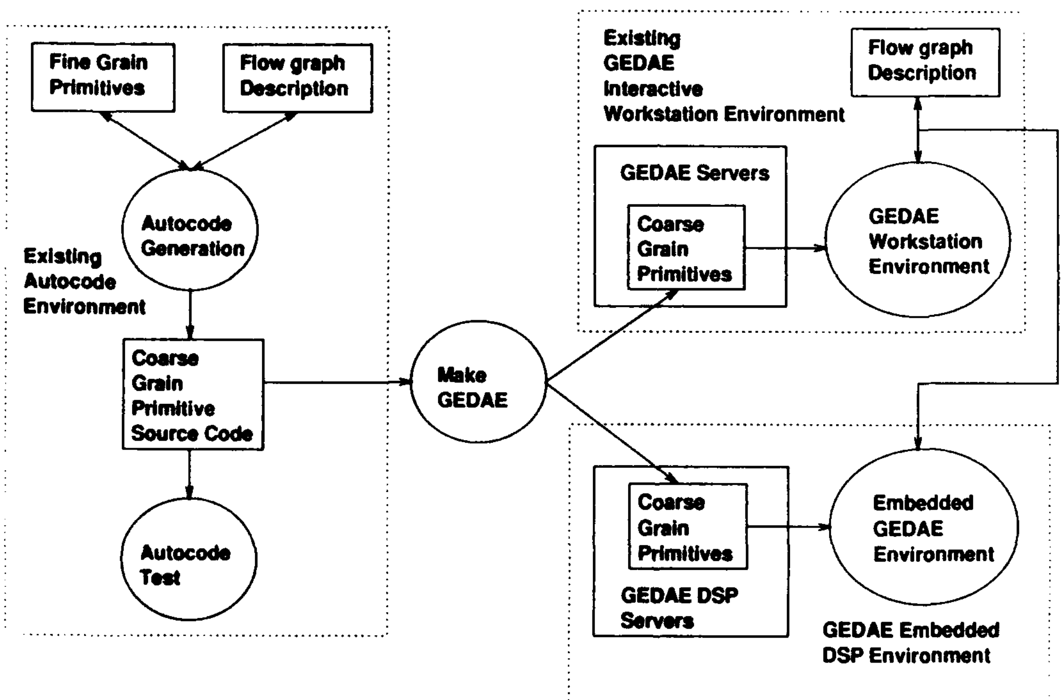


Figure 5-16. Proposed GE Distributed Application Environment.

Required extensions to these systems for RASSP are to create the ability to hierarchically build the coarse grain primitives (used by the interactive environment) from flowgraphs built out of the fine grain primitives (used in the autocode environment). A second requirement is to develop an embedded implementation that will achieve high efficiency and good memory utilization for parallel DSP arrays. A "Make" function will be developed to create servers for either the workstation or DSP environment. Thus, data flowgraphs tested in the interactive workstation environment can then be automatically embedded in a parallel DSP array. This provides a complete integrated system in which applications can be graphically developed and partitioned at the fine grain autocode level, executed and refined at the coarser grain workstation level, and efficiently implemented at the DSP level.

The RASSP Application Development Environment must support the integration of hierarchically linked algorithms onto networks of heterogeneous signal and data processors including, attached homogeneous parallel processors. It should separate the development of application-dependent modules, definition of the user interface, selection of processor/module assignments, description of data flow graph and the details of data flow implementation. The data flow details that are hidden from the user should include type casting, transfer across the network, continuous flow of data (for signal processing), reformatting for heterogeneous processors and data transfers to and from attached parallel processors.

- The system should provide the user with the ability to graphically create, edit, control, execute, and analyze hierarchical signal flow graphs.
- Provide the application programmer with the ability to create new fundamental function boxes and data type objects using a simple but complete set of tools.
- Hide all information from the user or application programmer that is not essential to his task. Knowing the processor on which a function box is implemented is unimportant to the user. An application programmer does not need to know the connections that form a higher level function from other primitive functions.

The DFG-based application environment must utilize many different types of signal flow graph functions for signal and image processing as well as scopes, scatter plots and image displays. The application programmer can supply new functionality at the primitive level, or the user can create it by building primitive functions into higher level functions using the hierarchical signal flow graph editor. Thus, after developing several applications, a large library of functions becomes available to address new applications.

5.5.3 Autocode Generation

The RASSP Software Development Environment must provide the ability to automatically generate target application code for multiple signal processors directly from the Data Flow Graph (DFG). The code must be generated using either HOL (C or ADA code), and must be able to accommodate libraries of optimized (assembly code) libraries to maximize performance and software reuse. This section describes current capabilities in autocode generation and extensions which must be made to meet RASSP requirements.

The RASSP autocode generation requirements are summarized in Figure 5-17, along with the current capabilities which exist in these areas. To date, autocode capabilities have been demonstrated that allow the user to create efficient very fine grain (sample by sample) data flow graphs with feedback and multiple clock rates, as well as scalar and multidimensional array data samples. Once a flow graph is created, code is generated, compiled, linked, and executed via a single menu click to provide an interactive test and validation environment. Control is provided from a panel that allows the code execution to be started, stopped and continued; this allows all parameters to be set (either before or during execution); and allows optional run time monitoring of all internal flow graph arcs. Support for multiple processors, however, is very limited. Some tools that currently support autocode generation are the Comdisco SPW tool, Mentor DSPstation, and GE's Distributed Application Environment. Complete descriptions of these tools are found in Section 10.

	Existing Capability
• Generate SDL Dscriptions from Data Flow Graphs	Limited
• Application Code	
– Synthesize HOL code from SDL	No
– Generate HOL and assembly code for each PE	Limited
– Extract and generate multiprocessor common code	No
– Support generation of control code—scheduling, synchronization, etc.	
· Centralized control	No
· Distributed control	No
– Support use of optimized libraries	Yes, All
• Support Code	
– Generate implementation-specific configuration files	No
– Generate code/symbol tables required by embedded OS	No
• Documentation Support Through Object Modules and/or Reverse Annotation	Limited
• Maximize SW Reuse Through Use of Libraries and Documentation via Back Annotation	Yes—Libraries No—Back Annotation

Figure 5-17. DFG-driven autocode generation requirements.

Many extensions to today's autocode capabilities are required on RASSP. The autocode capability should allow the user to create coarse grain primitive functions from fine grain autocode flowgraphs. In addition to the source code that normally gets generated to implement a particular function, code will be generated that encapsulates the function. The encapsulation routines will make use of facilities to allocate memory buffers, dynamically specify data flow requirements at run time, handle user settable parameters, and save and restore state information. Using this capability, a user will be able to develop efficient distributed applications using fine grain primitive flow graph elements with a minimum requirement for writing new code.

The autocode capability can be enhanced to allow code to be targeted to specific DSPs in order to gain maximum efficiency. This is done by allowing Primitive Function Descriptions in the autocode generation procedure to have both default C code associated with them and code for one or more specific DSPs. When building an embedded DSP server, the autocode capability is run on those flow graphs containing such retargetable primitives to produce code specifically for the given DSP. As a further means of enhancing the generated servers efficiency, the object code function libraries used by the "Make" program are specified for each DSP so that hand coded assembly routines are utilized when available. Thus, by providing optional overrides of the default libraries and primitives, highly efficient DSP code is created.

Another enhancement of the autocode capability will be to allow the expression of data parallelism. Data parallelism can be readily expressed in a data flow graph by connecting N dimensional matrix outputs to functions that operate on less than N dimensions. For example, a 2 dimensional image can be sent to a function that takes the square root of scalar quantities. The obvious interpretation of such a connection is that the square root is to be applied pointwise to each element of the image. The autocode capability will be enhanced to allow data parallelism to be expressed in this way and to generate code that correctly implements the desired functionality. This capability will increase the reusability of each primitive, avoiding the necessity of defining flow graph elements for every level of dimensionality used in the system. It will also allow parallelism to be explicitly expressed so that the embedded DSP mapping routines can take advantage of it.

Parallelizing an M dimensional function on an N dimensional matrix, as described above, is simple because the function to be parallelized is essentially stateless. An example of a stateful requirement for parallelization is high speed communication signal processing where the multiprocessor speed improvement available from pipelining the incoming data may not be sufficient to keep up with the real time data requirements. Parallelization of stateful functions require state information to be communicated from a processor working on data from an early time to the processor working on data at a later time. Further, this state information must be made available by the first processor before it is needed by the second processor in order to keep up with the real time input data rate. While this type of parallelism is readily expressible in a data flowgraph, there is no guarantee that the real time state communication constraints will be met. Either a timeline simulation needs to be done for the parallelized flow graph, or better, if the hardware is available, the user could automatically generate code for the run time DSP system, and using the embedded DFG environment to extract a functional timeline with the monitoring tools to see that the real time constraints are met. Integration of such functions with a timeline simulation capability will be performed so that flowgraph time constraints can be explored in the absence of the final embedded hardware environment.

Finally, automated code generation of support software for parallel processing elements must also be supported. From the DFG connectivity and mapping files, the tool must be able to: 1) automatically generate the information required by interprocessor communications routines, and 2) provide the support software (real time OS, etc.) with appropriate symbol table information to efficiently boot, test, and control the system. Such extensions have been demonstrated in pieces in a number of tools, but have yet to be integrated into a single comprehensive tool as described here. The RASSP development program will extend the existing commercial capability to provide the above requirements for RASSP.

Embedded Code Generation: An embedded DFG capability will make it possible to take the flowgraph algorithms developed and tested in the Interactive Workstation Environment and map the primitive flow graph functions to a high speed parallel DSP embedded system. The heart of the embedded system is the run time kernel which will implement the encapsulation library. [illegible] library will allow the same encapsulation code that was used to generate the workstation environment servers to also implement the embedded system servers. The library must be small to be usable in the memory poor environment of DSPs. The library must also be efficient to gain the throughput advantages provided by DSPs. As much of the current ability to control and observe running applications will be implemented as is consistent with these size and speed constraints.

5.5.4 CASE Support

The requirements for the software support CASE tools that will support RASSP are generally stated:

1. The software that is automatically generated by the individual modules of the DFG must be collected and organized into a complete and consistent group that allows the interfaces to be clearly described and defined.
2. The control software must be automatically generated and described to interface with a real time operating system for the DSPs.
3. Interfaces control and I/O from other Data processing components must be defined.

The major requirements for the RASSP CASE tools is that they support Reverse Engineering (RE) of the software. This results from the autocode generation inherent in a RASSP development methodology and the need to organize and interface to conventionally generated software. Given a autocoded software module and using the RE capability, the control and data structure of the module will be represented in a Structure Chart format (Constantine/ Yourdon) and captured in a consistent data base format consistent with the support services needed for the life cycle as described in Section 5.5.1.

Summary of Available CASE Tools: Surveys of CASE tools have appeared in many periodical publications. A comprehensive activity of the Software Technology Support Center (STSC) which operates out of the Ogden Air Logistics Center at Hill Air Force Base has produced an up to date survey (April 1992). The STSC Requirements Analysis & Design Tool Report shows a list of approximately 170 CASE products. When the RASSP requirements of CASE, real time, Workstation compatibility, and RE capability are applied to the list, the list can be narrowed to two primary vendors that produce a baseline CASE toolset. This forms a base that can be built upon to satisfy the RASSP requirements. Figure 5-18 compares the two vendors, both of whom have existing products that are already integrated into a CAD framework such as provided by Mentor Falcon or DAZIX Intergraphics.

The overall capability of a CASE tool set is summarized in Figure 5-19. This shows that the CASE set of tools contains a shared data repository of the functional designs, the data and the code. The CASE tools are integrated with a debugging environment that can be used to test the generated code. This capability is also integrated with document editors to produce the required documentation. The debugging environment capability is listed in Figure 5-20 and is part of the requirements of the RASSP software system. The RE function produces the various CASE descriptions of the code modules. It is important to realize that if additional control code is needed (which may not be autocode generated) this code may be easily created and integrated with the existing autocode. The key item that makes this process operate is that the forward and reverse process of code generation are synchronized. The Synchronization of the RE and the forward generated code provides another capability needed for RASSP as shown in Figure 5-21. As the application is being developed, it may be necessary for reasons of efficiency, omission, or required optimization to be able to insert specialized code segments. Even though the RASSP design should anticipate all the situations, in reality, provisions should be included to be able to make code changes. To take care of exceptional situations. However, the capability must exist to capture these changes as part of the design, and this incremental change capability must be integrated with the RE capability to ensure a consistent software design.

- Criteria for Selection of Case Tool Vendor
  - Integrate into Framework
  - Support Reverse Engineering Code Design
  - Support Complete Suite of Case Tools Not One Specialized Tool
  - Supports Large Software Projects > 100K Line of Code

Vendor/Tool	RASSP Application	Key Capability	Language
Integrated Development Environment/Software Through Pictures	• Improved Software Development Process • MIL Standard Documentation	• Synchronization of Code to Structure/Data Diagrams • Open Architecture to Include RTM, Code Debug, Documentation Tools	• C • ADA Partial • C++ Partial
CADRE/Teamwork Modules RT, SA, SD, IM, IPSE, ASG, ASB, ASG, ASE, OOA, OOD, DB, C Rev	• Separate Modules Perform Overall Task From Structure Analysis to Documentation	• Structure Analysis with RT Extensions • Reverse Engineering in C • Documentation Extraction • Consistent Data Formats	• ADA • C Partial • C++ Partial

Figure 5-18. CASE tools.

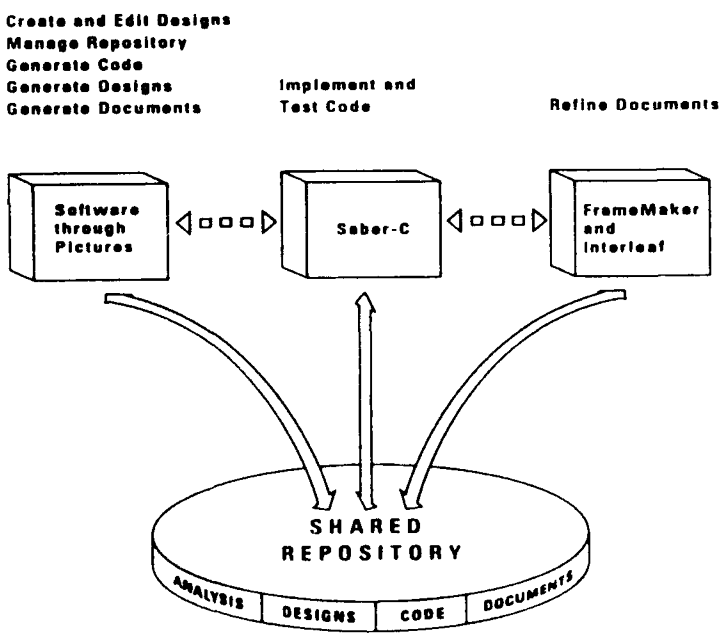


Figure 5-19. C development environment.

Saber-C

- **Error Detection**
  - — Static (Incl. Cross Module)
  - — Run-time
  - — Customization
- **Debugging**
  - — Extensible Breakpoints/Watchpoints
  - — Complete C Language (Macros)
  - — Debug Code Fragments
- **Browsing Tools**
  - — Function Call Trees
  - — Data Structures and Links
  - — Error Messages and Locations
- **Interactive Workspace**
- **Incremental Linking**
- **Interface for X Windows**

Figure 5-20. Capability of debugging environment.

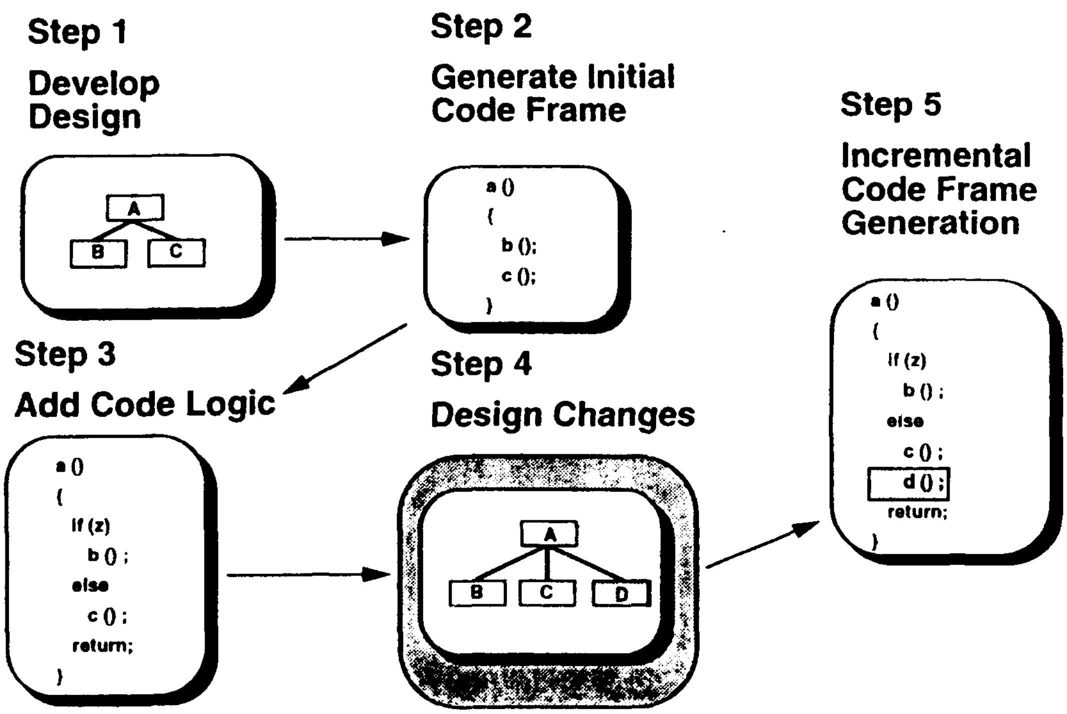


Figure 5-21. Incremental development preserves previous work.

Required Development for RASSP: As described above a baseline CASE tool capability for RASSP exists. There are three areas that require development to enhance the current tool sets.

First, the selected CASE tool set need to be integrated into the RASSP framework and the integration of the shared repository of data or data base into a consistent set. The data that is presently deposited into the data base is a result of the Forward and Reverse process of code generation. The overall system design is handled by the DFG design methodology. The design process must to be able to incorporate data from the system and DFG part of the design phase into the software documentation and descriptions. This can be directly added to the software data base when appropriate interfaces to the data base are established.

Second, the CASE tools that have been developed primarily support a collection of single processors operating in a loosely coupled environment. The RASSP requirements include the generation of software for tightly coupled parallel processors. When the autocode is generated and allocated to specific data processors, the CASE tools must be capable of reflecting the allocation and documenting the parallel processing function. Additionally, the debugging capability must operate in a parallel environment. In order to handle this parallel environment the CASE tool must deal with multiple modules operating on different DSP processors. The data base must integrate all the system data structure and assure consistency over the multiple processors. This include the handling and description the data structure as they are allocated to the shared and local memory.

Third, the ASIC part of the design must be interface to the DSP software that is managed by the CASE tools. As the ASIC hardware interacts with the DSP functions and data specialized I/O modules and capability must be incorporated into the tool to handle this interface. There is a need to develop specialized interface to handle the operating system software that supports the parallel environment. Since the operating system software is an integral part of the RASSP system, this software must be integrated into the CASE tools and the designer should be able to invoke this functionality as part of the CASE tools.

6. RASSP UTILIZATION/ACCEPTANCE ISSUES

Long term success of RASSP is dependent on widespread acceptance and utilization in the design community, and support by the EDA CAD vendors, and the DSP technology development organizations.

GE Aerospace has established a plan for its Engineering Process Improvement Program for implementing and supporting a design system that will grow as tools mature. This strategy provides a natural handoff for the RASSP program to integrate into the standard design system of a large aerospace firm. Also a number of aerospace companies (Rockwell, etc.) have plans for developing similar design systems and have expressed a strong interest in RASSP. This is driven out of a recognition of the need to reduce engineering and production costs for large aerospace companies to remain competitive in the projected defense business. EDA vendors, facing declining markets over the last two years, are also interested in expanding their markets through involvement in new developments like RASSP.

The GE RASSP team will provide annual briefings and demonstration projects to support the dissemination of RASSP to industry.

This section details some of the key deterrents to RASSP utilization, and, and possible approaches to overcome these and achieve acceptance. In addition, a discussion of usability and technology obsolescence as issues which RASSP must address for long term success is provided.

6.1 RASSP Deterrents and Enablers for Success

User Community

Deterrent: DoD/Aerospace companies have installed or evolving concepts for platform and processor design, use of standards, and logistic support that are based on large investment, and legacy design practices. The current apparatus (program approaches, and design methodologies) used by SPO's and Aerospace companies are not easily changed. Promises have been made previously on advanced technology concepts, and it took many years for the R&D community to deliver on their promises.

Enablers for Success: The user community, services, SPO's, and standards groups need to be involved up front in the program planning phases, when key decisions are made. The program needs to find advocates in high level DoD and Congress. Successful demonstrations of the technology early in the program are also critical for gaining program support.

Applications

Deterrents: Each company has product groups responsible for a given application area, and has installed approaches for all design levels from systems design through life cycle cost management. New approaches to implementation, such as the model year concept, may be viewed by designers as forcing compromises in performance and flexibility.

Enablers for Success: Provide early design examples and demonstrations that highlight flexibility and good achievable performance. Demonstrate application of RASSP to several design styles and system types.

Infrastructure

Deterrents: Enterprise design system infrastructures already exist at each industrial organization. Capture of existing design information, and commercial databases into the RASSP system for reuse will be difficult at early stages in the RASSP, making system utilization unattractive. Vendors of relevant DSP technology will likely resist sharing proprietary information on planned product releases with the RASSP system developers and users.

Enablers for Success: Library management / modeling concepts that have utility to all programs, and can utilize existing libraries / databases are key to acceptance. The RASSP program must establish alliances with the DSP vendors, and develop mutually beneficial approach for dissemination of pre-release design information.

RASSP Design Methodology

Deterrents: Each company has existing design methodologies, used for each specific product area. There will likely be a perception that a RASSP design methodology will result in increased cost in the design process (as opposed to savings).

Enablers for Success: Early productivity benchmarks need to be performed and the results demonstrated. The productivity parameters of ongoing design methodologies must be measured, for comparison to productivity parameters associated with the RASSP design methodology and design system. The measured benefits of program efforts similar in concept to RASSP (such as the GE-EPI) need to be made available to potential RASSP users.

RASSP Framework

Deterrents: Companies already have a design systems in place, hence have significant investment in this area. RASSP will likely require new acquisitions of hardware, software and training.

Enablers for Success: The RASSP framework team will work ongoing efforts to establish standards in the framework area, and new developments will be required to conform to those adopted standards (ex. CFI). The RASSP team should leverage the priorities of the commercial EDA industry to the maximum extent feasible.

Manufacturing

Deterrents: Each company already has a defined manufacturing facility with capital and existing automation approaches. RASSP automated manufacturing approaches will require new capital and training. Investment in new manufacturing approaches will be hard to sell, based on DoD low volume production predictions.

6.2 Discussion of Issues for RASSP Long Term Utilization and Success

Usability Issue: RASSP is a technology application, and over the long term its designers and builders it cannot be expected to provide active direction and hand-holding to potential users to the extent that they will be capable of providing in the early stages of deployment. Hence, in the long term, users will be heavily influenced by usability considerations in their perception of RASSP's utility. There are many factors that can significantly improve ease of use. First, ease of use is enhanced by graphical user interfaces (GUI) to all tools in the environment. RASSP will use GUI extensively throughout the environment, and will strive to maintain some measure of uniformity in the way GUI is implemented in each tool. Secondly, ease of use will be enhanced by adhering to industry standard data formats, and by providing transparent translation utilities between different formats. Third, RASSP must provide a sharp focus from among its many capabilities. The idea is that RASSP may be capable of doing many things, but the one thing it does extremely well is the model year methodology. RASSP's model year methodology will be presented to the user as a clearly defined set of steps that are intuitive, well supported, and that provide obvious benefits if followed.

Technology Obsolescence Issue: To guard against technology obsolescence, RASSP will be implemented as an open system, whereby new tools can be integrated into the system through standard interfaces. Tool integration into RASSP can be viewed in two contexts: the physical aspect and the methodological aspect. The physical aspect of integration concerns communication between the new tool and the rest of RASSP, utilities and formats for data interchange, and extending the common aspects of the GUI to the new tool. These requirements will be well supported in RASSP. The methodological aspect of tool integration concerns how the new tool or technology can be brought into the model year methodology, and what changes would result in the methodology itself.

Digital hardware design tools and technology will face the technology obsolescence problem in "back end" (synthesis, design for testability, partitioning) tools rather than in the "front end" graphical design tools. The use of VHDL as the common medium of expressing design and test data between the tools will ease the burden of introducing new "back end" design tools.The design of RASSP will ensure that tool interfaces will be standardized.

In particular, there will be no private data formats used to convey data between tools, thus overcoming one of the major problems in introducing new tools into the environment.

7. MANUFACTURING

7.1 Design Center/Manufacturing Electronic Interface Approaches

A CAx design environment (system-module-chip) for the signal processor domain does not exist which links technologies and manufacturing knowledge throughout the entire design and manufacture cycles. Lack of this environment has caused poor product designs and delays in fielding, resulting in higher costs. There is great potential for reducing design time and facilitating the acceptance of RASSP modules in complex systems if integrated design tools were available that linked directly with the multiple RASSP foundries.

For the RASSP study, multichip module foundries (especially those aspiring to become ASEM[1] foundries) are targeted for analysis for meeting RASSP requirements. This section reviews the status of MCM merchant foundries and describes the need to develop and implement foundry interfaces between the CAD environment, manufacturers and suppliers.

It is important that all design tools and electronic links be developed in harmony with the hardware enabling technologies and have compatible interfaces and standards so that they emerge simultaneously, ready for use by designers, manufacturers, and suppliers.

It was determined that such interfaces will eventually need to be implemented electronically to support the goal of rapid prototyping. Electronic linkage between design centers, suppliers, and manufacturers (enterprise integration) would be required to achieve this goal. This would require close coordination with other DARPA funded activities under the ASEM program, other DoD programs, and with other efforts underway by standards groups.

Recommendation: It is recommended that DARPA consider the use of the existing ASEM CAD/CAE/CAT/CAM Interface Specification Alliance to ensure that RASSP domain specific interfaces are also defined and submitted as standards. This will ensure industrial acceptance and transition in to business practice. Major advantages and outputs will be:

- Unification of ASEM and RASSP participants for leverage and compatibility
- Approach will produce comprehensive detailed descriptions of RASSP design tool interfaces in a format which promotes and accelerates their wide dissemination and realization into commercial and military business practices.
- The EXPRESS information-modeling language will be used to describe the exact content of RASSP design information and data exchange interfaces as a STEP application protocol , making the interface specifications readily adaptable by EDA vendors, RASSP vendors, and suppliers (essential for model year concept).

[1] ASEM (Application Specific Electronic Modules)

- Standards experts will be an integral part of design activity modeling to accelerate standards adaptation and reduction of duplicated efforts.

7.1.1 Unified Alliance Standards Activities

The RASSP program must incorporate standardization experts from the beginning of the effort, who, by working closely with the technical specialists within the Unified ASEM/RASSP Alliance, will accelerate the definition of industrially acceptable interface specifications and their compatibility with existing and emerging standards. The ATLAS Standards Lab at MCC is ideally suited to assist in defining the interface specifications in a formalized procedure to ensure their compatibility with the International Standards Organization (ISO) Standard for the Exchange of Product Model Data (STEP[2]). The EXPRESS[3] information-modeling language or similar language should be used to describe the exact content of RASSP domain specific design information and data exchange interfaces as a STEP application protocol[4], making the interface specifications readily adaptable by EDA vendors, RASSP/ASEM vendors, and suppliers.

This interface definition activity provides an opportunity for industry to establish a common interface format before any major investments have been made in design tool developments and manufacturing equipment. This will result in significant cost savings by avoiding later adaptations by industry to comply with standards after the fact.

The objective of this recommended RASSP program effort is to determine which interface specifications are candidates for formal standards adaptation in a sequence as shown in Figure 7-1. The Unified ASEM/RASS CAx Alliance must work with national and international standards bodies to prepare and submit any candidate RASSP CAx interface standards which has resulted from the efforts. The interface specification models must be compiled for publication.

It is recommended that the RASSP program employ experts in the STEP standardization methodology to assist in the development of the Application Resource Model (ARM) document (it must include the EXPRESS information model plus an activity model and other information pertinent to the technology of the product) and to employ these experts to accelerate the movement of that ARM through the various committees within ISO TC184/SC4 which are involved in the STEP process. The output of this recommended standardization effort is the RASSP Standards Submissions to Appropriate Standards Group, the analysis of the RASSP design

---

[2] The STEP standard (International Standards Organization 10303) is intended to provide computer-interoperable information models for representing the product data necessary and sufficient for product-data collection, storage, and transfer as a means of standardizing the commercial transactions associated with products covered by the 10303 standard.

[3] EXPRESS is an information modeling language developed for product data exchange model definition. First developed under USAF funded PDDI program, EXPRESS is a computer processible, object oriented textual language capable of modeling things and relationships, algorithms, data structures, and graphical forms.

[4] Application Protocol (AP) defines the scope, activity, and information domain of an application and specifies the rules for using VHDL, IGES, EDIF, or some other standard to enable the transfer of the application information. Information models are defined in the EXPRESS language.

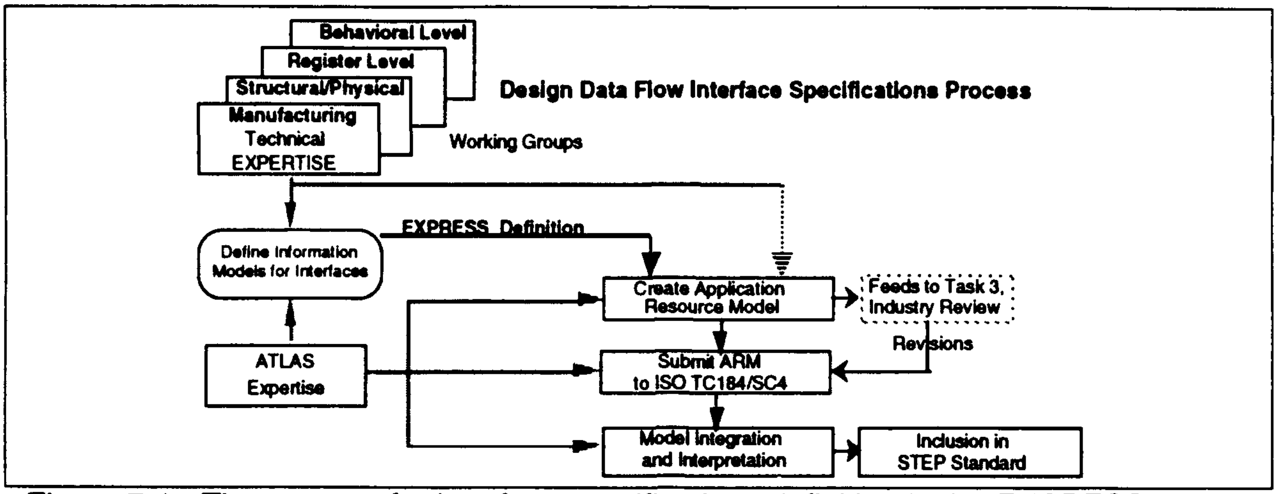


Figure 7-1. The process for interface specifications definition in the EXPRESS information modeling language and transitions through industrial review and submissions as candidate standards. Such effort is required in the RASSP program to capture standard candidates specific to the signal.

environment will lead to defining standards. Interface specification models will be submitted as a candidate standards for acceptance by the appropriate standards group, i.e., International Standards Organization (ISO) TC184/SC4 STEP Standard and the CFI CIR-TSC Electronic Databook (EDB).

7.1.2 Electronic Linking

The results of the interface specification definition developed by the Unified ASEM/RASSP CAx Alliance is needed to support the implementation of an electronic link between design centers, suppliers and manufacturers/foundries. The interface specifications will support the development of an industrial interface standard for linking multiple CAD frameworks with the manufacturers. The desired result is illustrated in Figure 7-2.

This foundry interface is unique in that it is a critical link between the CAD/CAE design environment and manufacturing. This is probably one of the weakest and poorest defined interfaces because of the technology process and material dependent requirements which exists for the various electronic manufacturing foundries and lack of maturity. The RASSP program must create interface specifications which make technology dependent requirements readily available to the RASSP designer. The designer must have technology specific characteristics for performance modeling as early in the design cycle as possible. Accordingly, the challenge is identifying and defining a complete set of interface specifications which describes this bi-directional coupling of RASSP "domain specific" design data information which can be electronically accessed over a network.

For effectively coupling of CAD systems with RASSP foundries, and for improving productivity with the foundry, the ideal situation described below and illustrated in Figure 7-3 and 7-4 needs to be implemented.

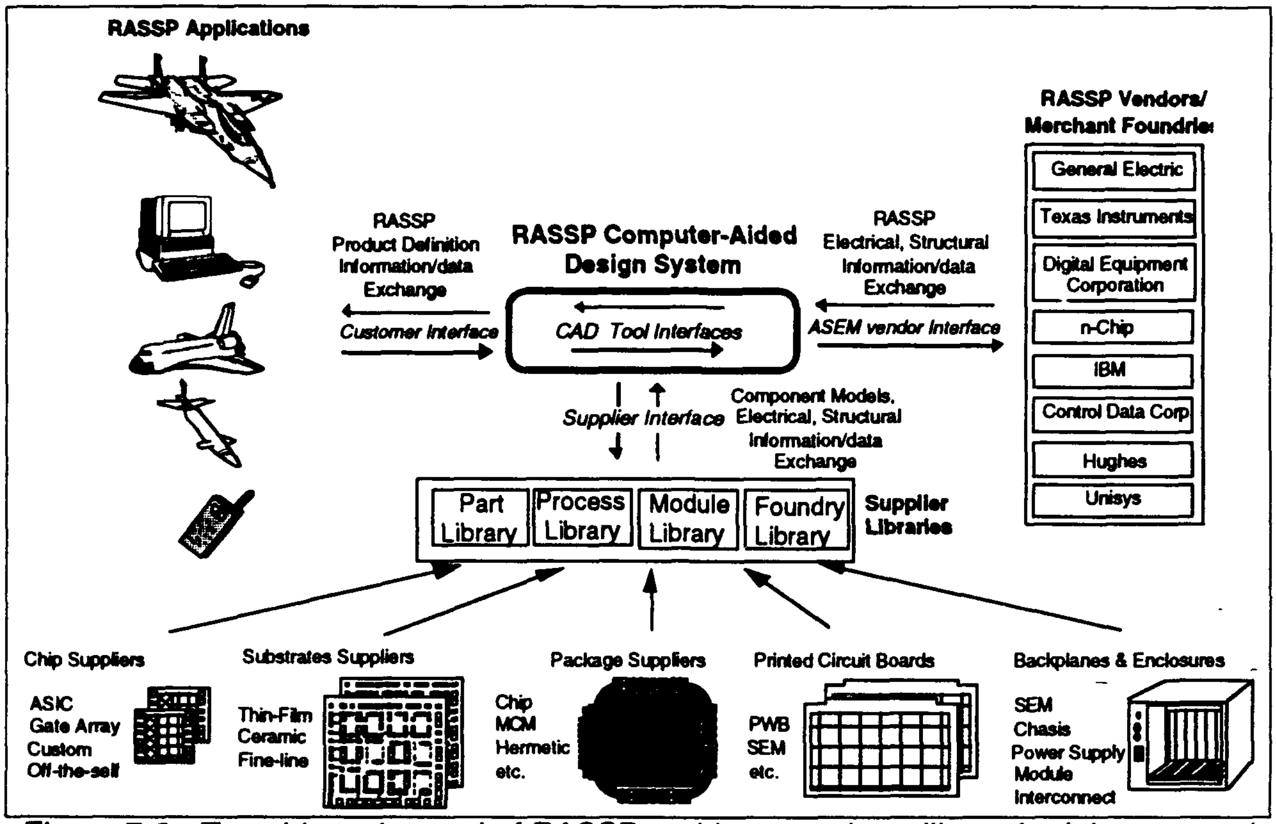


Figure 7-2. To achieve the goal of RASSP rapid prototyping will required the eventual electronic linking between design centers, RASSP foundries and suppliers. This is needed to enable the concurrent engineering required to achieve the first pass success for RASSP systems/modules.

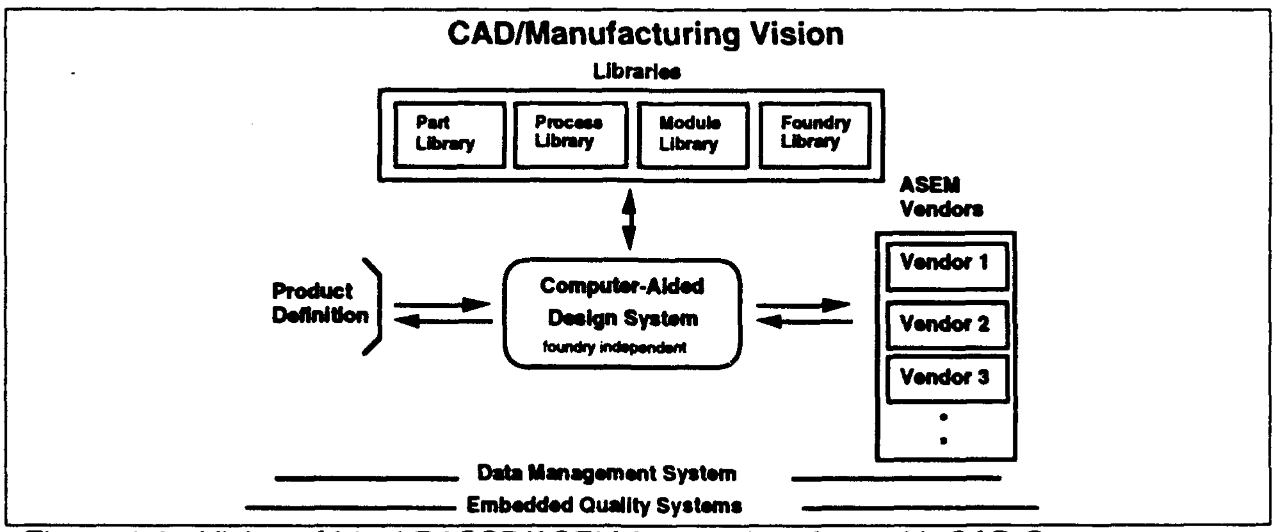


Figure 7-3. Vision of Ideal RASSP/ASEM foundry Interface with CAD Systems

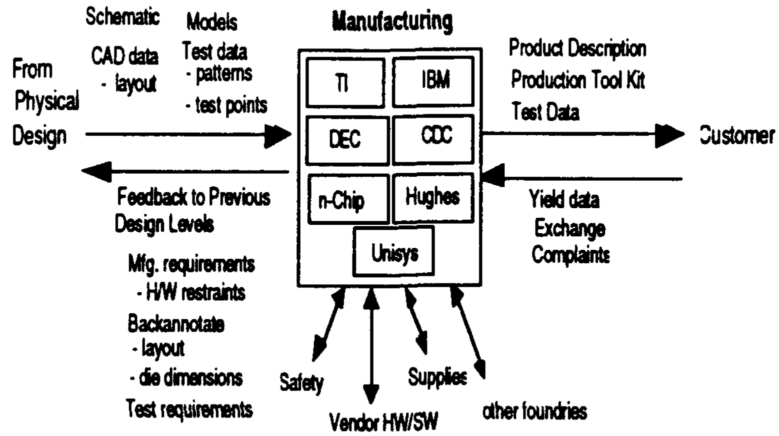


Figure 7-4. Examples of interfaces required to be defined specifically for RASSP foundries. The ASEM CAx Alliance will be covering these same issues and can work specific issues for RASSP simultaneously. Information can be shared openly with other organizations contracted to develop design tools and high level description languages, for example.

- Commercial CAD systems must interface with "all" RASSP foundries.
- Standard formats are needed for bi-directional data exchange between design and layout, design and test, and layout and manufacturing.
- Frameworks standards are needed that make required tools accessible to users of already installed automation systems, perhaps through a common object oriented data base.
- A data management system for product design data is needed that provides configured data to all software tools supporting the product life cycle.
- A data management system for manufacturing build data is needed for use within the foundry that provides data in a form usable by installed hardware and supports collection and analysis of manufacturing data for process improvement.
- Quality concepts are built into the design and data management systems and are fed by rulebases backannotated from manufacturing.
- Design-for-test concepts are built into the design and data management systems and are fed by rulebases backannotated from manufacturing.

The RASSP program must bring together foundry personnel who will define the necessary enhancements specific to the RASSP domain and forward them to appropriate standards and ad hoc committees for implementation.

The cost effective, rapid prototyping of RASSP systems will require electronic linking of design centers, foundries and suppliers. The implementation of the electronic link must be done on a network service that will support true electronic commerce. DARPA should support the implementation and demonstration of an electronic network that provides the following capability:

- CAD conferencing
- Electronic databooks
- Directory services (hierarchical in nature for ease of navigation)
- Security services to support encryption based user authentication and access control
- Advanced E-mail services to enable private/secure communication through text, video, and audio.
- Remittance services to enable companies to complete business transactions through the electronic remittance of funds.
- Compatible with all major workstations and PCs.
- Supports dual protocol for Transmission Control Protocol/Internet Protocol (TCP/IP) and Open Systems Interconnection standards (OSI)

The capabilities sited above will ensure the longevity of the RASSP program and is required to support the model year and rapid prototyping concept. A phased implementation is recommended to demonstrate and benchmark the electronic linking to multiple foundries and suppliers. The eventual and required electronic network system will appear as shown in Figure 7-5.

7.1.3 Status of MCM Merchant Foundries

A survey of companies, which are positioning themselves to become merchant foundries for ASEM (Application Specific Electronic Modules) and RASSP modules/systems, was conducted by the Microelectronics and Computer Technology Corp. (MCC) during the RASSP study phase. The purpose of this survey was to obtain a snap shot of what capabilities and within what time frame they expected to become available as merchant foundries. The survey also obtained information on their CAD tools, test ability, and areas in which equipment development is still required to enable the cost effective manufacture of ASEM/RASSP modules/systems. Since all of these companies are also part of the DARPA ASEM CAD/CAE/CAT/CAM Interface Specification Alliance, they all agreed to participate in the survey and to cooperate in follow-on programs which would help implement the cost effective design and fielding of RASSP modules/systems. This Alliance is shown in Figure 7-6.

It is recommended that the Alliance be utilized in the RASSP program to accelerate the integration of design tools needed for implementing a realistic, cost effective, rapid prototyping system design environment by identifying and defining the design information and data interface specifications and applicable standards needed throughout all levels of the design-to-manufacture cycles. It is further recommended that these interfaces be implemented in a program that requires participation of multiple major EDA vendors and ASEM/RASSP manufacturers.

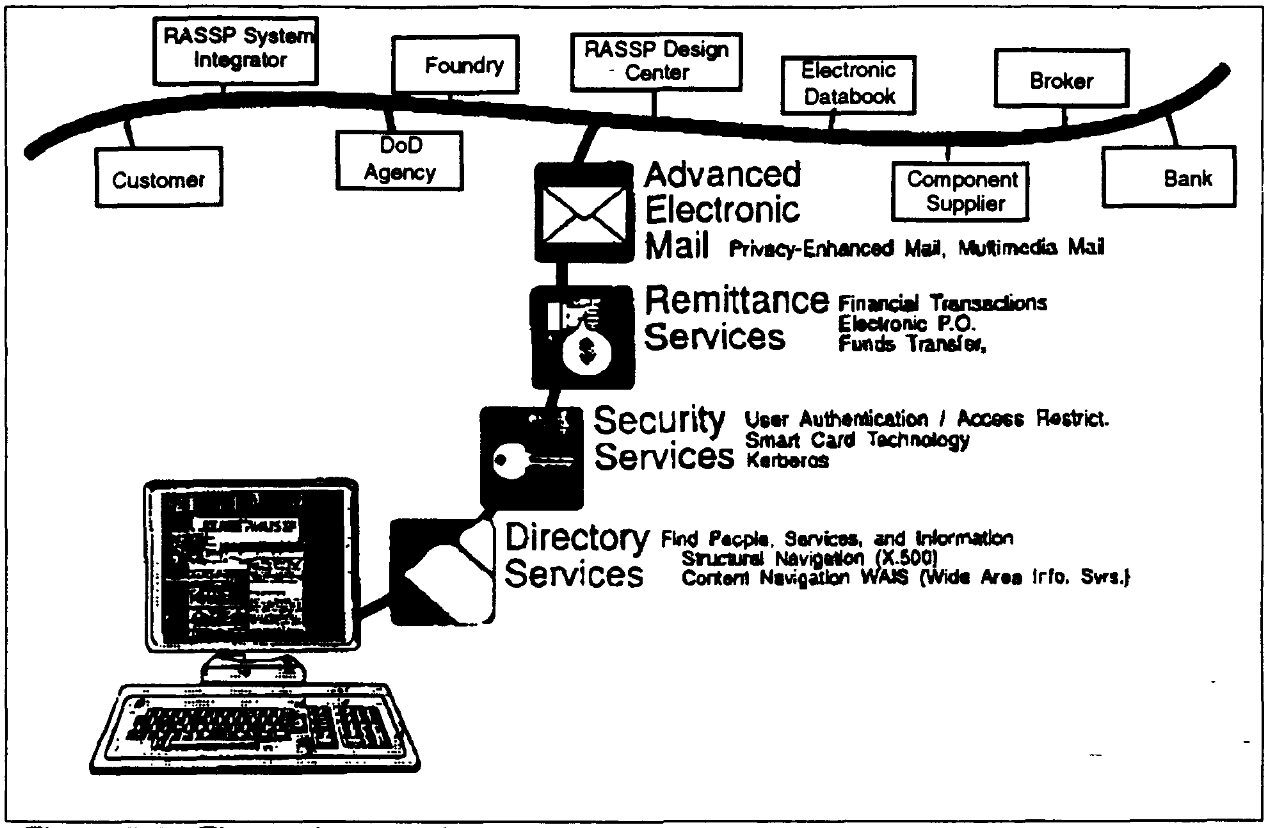


Figure 7-5. Electronic network system.

The specific objective or task for the Alliance will be to develop a design activity model for RASSP systems in a unified collaborative effort by industry which will enable the standardization of design information, data and foundry interfaces to achieve more than 50% reduction in RASSP design time and costs. The output must be compliant with STEP[5] standards.

Figure 7-7 lists the major types of interfaces which may require RASSP domain specific specifications. Based on the survey and previous efforts of ASEM CAx Alliance organizations, defining the interface between the CAD/CAE environment and the manufacturer is critical for achieving rapid prototyping and the eventual electronic linking. The design productivity improvement which could be gained by defining and implementing the interface specifications illustrated in Figure 7-7 could cut in half the design time of RASSP modules. The potential productivity improvement is illustrated in Figure 7-8.

Standardization and definition of interfaces could greatly improve the efficiency of MCM design if EDA vendors, end users, and MCM foundries, have a common understanding of design flow, entry points into manufacture, manufacturing interface

[5] The STEP standard (International Standards Organization 10303) is intended to provide computer-interoperable information models for representing the product data necessary and sufficient for product-data collection, storage, and transfer as a means of standardizing the commercial transactions associated with products covered by the 10303 standard.

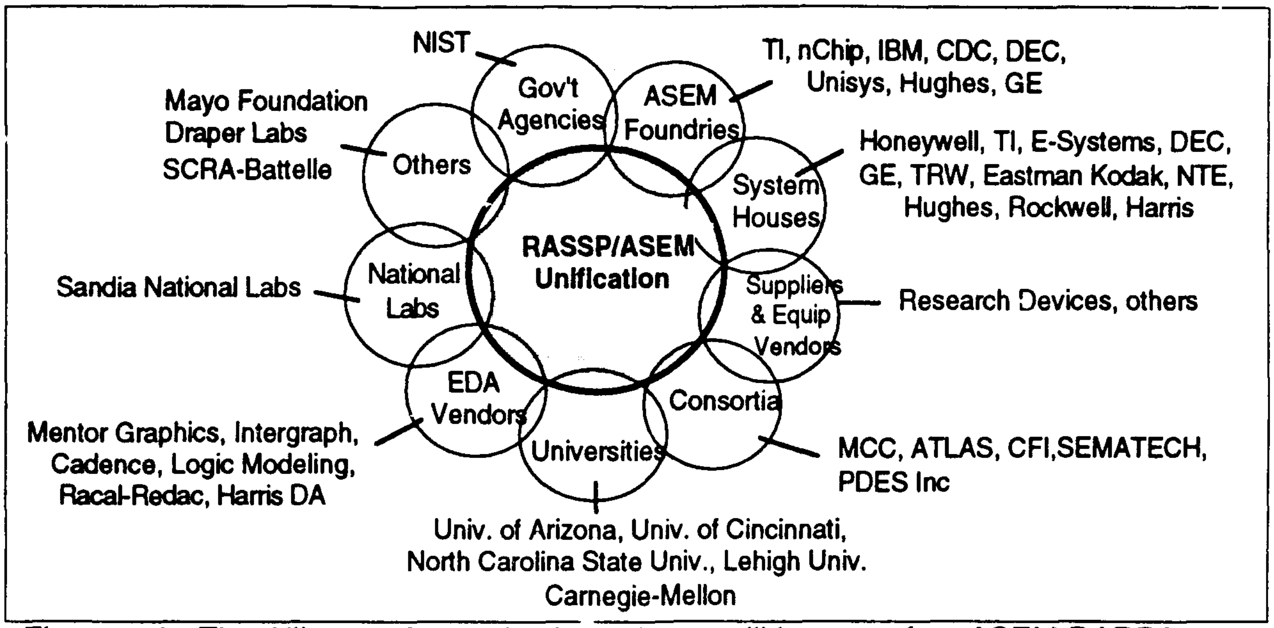


Figure 7-6. The Alliance of organizations shown will be part of an ASEM DARPA contract to define the interface specifications for the ASEM CAD/CAE/CAT/CAM design environment. The RASSP program can leverage from this existing Alliance to incorporate RASSP domain specific interface issues and model year concept standardization. The results are targeted for ease of adaptation by industry into business practice. This Alliance will be managed by the Microelectronics and Computer Technology Corp. (MCC).

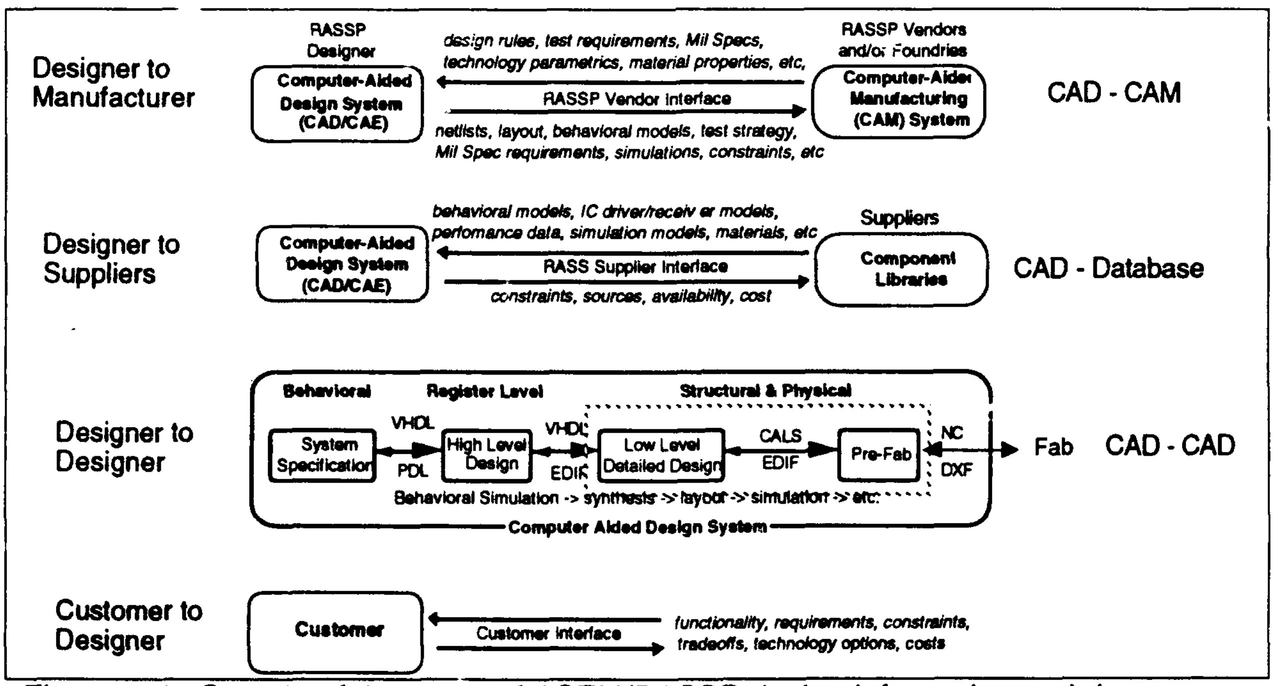


Figure 7-7. Sample of the types of ASEM/RASSP design information and data exchange interfaces to be defined by the Alliance. The unification of the Alliance to also address the needs of RASSP will provide significant leverage to the RASSP program.

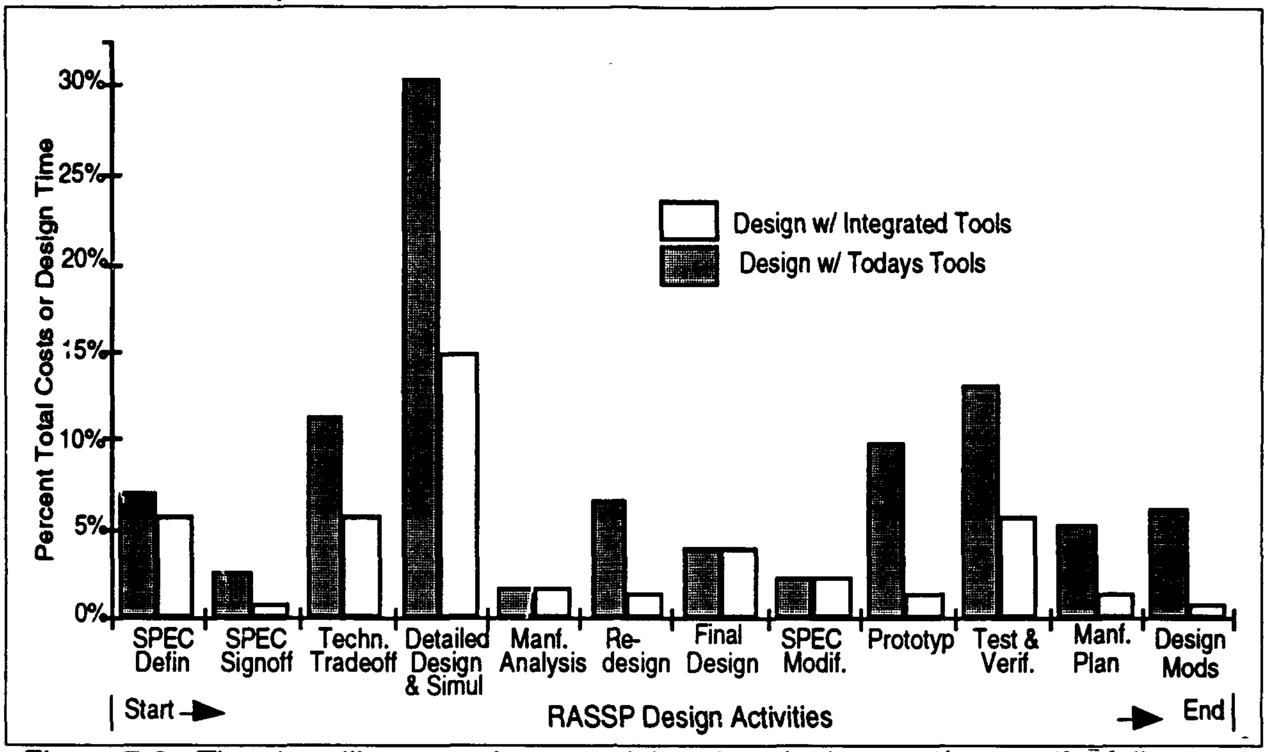


Figure 7-8. The chart illustrates the potential savings in time and/or cost if a fully integrated RASSP design environment existed which linked with manufacturing design information.

requirements, and, of course, real world problems that designers see as bottlenecks, gaps, or shortfalls. The unified ASEM/RASSP Alliance would help to ensure that the RASSP program meets the real needs of industry and DARPA in rapid prototyping of RASSP modules and systems.

7.1.3.1 Foundry Survey

The results of the survey are presented in Tables 7-1a, b, and c. For the study phase of RASSP, a brief survey was conducted to establish contact with all the potential RASSP foundries. A more detailed survey was not conducted because this would duplicate the work which is planned for the ASEM CAx Interface Specification Alliance. The results of any ASEM efforts will be shared with any future RASSP program developments.

7.1.4 Enterprise Site Interface to Automated Manufacturing Center

The RAMP FCIM automated manufacturing technology architecture supports modularity and flexibility. This architecture can be tailored to support any printed wiring assembly design requirements. This architecture can also be tailored to support multiple factory floor models. This architecture is designed to be integrated into any existing electronic design and manufacturing enterprise. To accomplish this RASSP integration the enterprise interface requirements have been established and can be tailored to meet specific manufacturing requirements.

Table 7-1a. MCM foundry survey status results.

	Hughes	GE	TI
Status to Manufacture	Today	Today for Prototypes	Today for Prototypes 4th Q 92 for Volume
Interconnect Technologies	LTCC Thin Film (Al-PI)	Thin film (HDI)	Thin film (HDI)
Support Digital Mixed Signals Clock Speeds	 yes yes >500 MHz	 yes yes >1 GHz	 yes yes >1 GHz
Production Capability Prototypes Medium Large	 yes yes yes	 yes	 yes 4th Q 92 4th Q 92
Subassembly Capability	yes	yes	yes
CAD-CAM Link	fab	yes	yes
Future CAM	shop floor with config mgmt		Tool integration w/ CAD
Test Strategy MCM Test Mixed Analog digital Built-in Test Test Equip.	 BITE yes yes IMS XLII ATS 2 Univ Analog Stat	 Mfg defect test & final assembly yes yes	 Mfg defect test & final assembly yes yes HP 82000 VXI Instr.
Equipment Which Needs Development to Improve Cost/ Reliability/etc			• High rate metal dep. • low cost, high speed probes • high speed test w/ diagnostic • low cost/large area litho • high rate dielectric application
CAD/CAE Frameworks	Mentor Graphics Hybrid Station MCM Station Chip Grph Cadence Dracula Autocad Dsgn Workshop		? Harris Finesse
CAD Conferencing Capability	no		Yes (Internet)
Plan Electronic Linking w/ External Customers	yes		Yes

Table 7-1b Foundry status survey results.

	nChip	Motorola	Unisys	Martin Marietta
Status to Manufacture	Today	Prototype only Low volume 2 Q 93	Today Prototype volumes	Awaiting response
Interconnect Technologies	Thin-film deposited	Chip & Wire	MCM - L MCM - C assemb & Design	
Support Digital Mixed Signals Clock Speeds	 yes yes >100 MHz	 yes yes > 50 MHz	 yes no >250 MHz	
Production Capability Prototypes Medium Large	 yes yes yes	 yes 2 Q 93 low vol production 	 yes ? ?	
Subassembly Capability	yes	yes	yes (digital)	
CAD-CAM Link	Some process steps Bare substrate test data	yes (assemb equip)	no	
Future CAM	Wirebond data die placement	Full integration planned	TBD	
Test Strategy MCM Test Mixed Analog digital Built-in Test Test Equip.	 Substrates Module test per design JTAG when available Yes, as per design requirements No, only by design of customer General VLSI tester	 Characterization Burn in full functional Yes Yes - BIST and BS HP	 Full continuity substr Full functional for assemb no yes Internally developed for substrates Sentry for functional	
Equipment Which Needs Development to Improve Cost/ Reliability/etc	• Wirebond equip enhanc • Die attach equip • bare die tester • enhanc substr tester • better fixturing • probe cards for at speed testing	Parametric and characterization equip	Bare die testers	
CAD/CAE Frameworks	Mentor Graphics MCM station Cadence Allegro Edge IC tools	Cadence Allegro	Mentor Graphics Hybrid Station MCM Station Chipgraph	
CAD Conferencing Capability	no	no	no	
Plan Electronic Linking w/ External Customers	yes	yes	yes	

Table 7-1c Foundry status survey results.

	IBM	MCC	CDI	MMS	AT&T
Status to Manufacture	2nd Q 93 MCM-D Today for MCM-C	Today	Awaiting Response	Awaiting Response	Awaiting Response
Interconnect Technologies	MCM -D, MCM-C TAB, FC & wirebond	MCM-L, MCM-D wirebond,TAB, FC			
Support Digital Mixed Signals Clock Speeds	 Yes Yes	 Yes yes 300 MHz			
Production Capability Prototypes Medium Large	 Yes Yes Yes	 Yes			
Subassembly Capability	Yes	Yes			
CAD-CAM Link	Some	no			
Future CAM	Yes	yes			
Test Strategy MCM Test Mixed Analog digital Built-in Test Test Equip.	 GSSI, Test vector ? Yes, scan	 substrate connectivity Manual Yes			
Equipment Which Needs Development to Improve Cost/ Reliability/etc		Test equip to support at speed built in self test (analog and digital			
CAD/CAE Frameworks	Cadence - Allegro	Cadence Allegro Mentor Graphics MCM station Intergraph Harris Finesse			
CAD Conferencing Capability	no	yes			
Plan Electronic Linking w/ External Customers	yes	yes			

The Enterprise Site Interface module within the RAMP FCIM architecture receives and sends messages to and from the RASSP enterprise support activities. The Site Interface module converts the data elements associated with each message it receives or transmits to the corresponding data element required by the RAMP FCIM system or by RASSP. The RASSP interface activities transmitting and receiving messages are; 1) Production Planning and Control (PP&C), 2) Quality Assurance, 3) Maintenance, 4) Accounting, 5) Receiving, 6) PP&C (Material Management), 7) Packing and Shipping, 8) PP&C (Requisitioning), and 9) RASSP Engineering Design and Development

Branch (ED&DB). To integrate the RAMP FCIM system into the existing RASSP operational environment the following interface requirements information is provided.

7.1.4.1 <u>Interface Message Requirements Between the Site and RAMP FCIM</u>

The following paragraphs provide a description of the messages to and from RAMP FCIM and discuss how the messages may be used by the RASSP support activities.

7.1.4.1.1 Order Message Traffic Through RAMP FCIM

The following paragraphs provide information for starting an Order message for fabrication and assembly of requested items in the RAMP FCIM FFMs.

7.1.4.1.2 Technical Data Message

The first message that RAMP FCIM should receive is the Technical Data message from ED&DB which may be transmitted electronically or sent to RAMP FCIM physically either on tape or on an optical disk. When received, the RAMP FCIM system will validate the technical data to ensure that it is complete. If there is a problem with the technical data, the RAMP FCIM system will send a Technical Data Reject message to the ED&DB and to PP&C.

7.1.4.1.3 Order Message

The second message that should be received by the RAMP FCIM system is the Order message. RAMP FCIM will validate the Order message for completeness and accuracy before the message is accepted. If there is a problem with the Order message, the RAMP FCIM system will send an Order Status message to PP&C specifying the problem. PP&C will correct and resend the Order message to RAMP FCIM. Upon acceptance of the Order message RAMP FCIM will select the correct Manufacturing Engineering cell within the RAMP FCIM required to support the FFM(s) needed to fabricate and/or assemble the requested item(s).

7.1.4.1.4 Item Requisition Message

After acceptance of the Order message, the RAMP FCIM system will start the Macro process planning function and will develop an item requisition for each item noted on the bill of material. The RAMP FCIM system will send an Item Requisition message, for each item required to complete the order, to PP&C (Requisitioning).

The Item Requisition message will contain unambiguous commercial-off-the-shelf component procurement information along with the quantity required and delivery information. The message will also contain suitable substitute information if it is contained within the technical data.

The RAMP FCIM manufacturing engineer supporting the specific FFM determines the make or buy criteria for an item. He also determines if the item can be made within the RASSP shops or outside of RASSP. If determination is made to make the item within RASSP, the manufacturing engineer generates a work order and sets a flag in the Item

Requisition message reflecting a work order number. If it is determined that the item must be made outside RASSP the manufacturing engineer will use the Item Requisition message to requisition the item. This message will also contain all technical data required to obtain the item.

The RAMP FCIM will coordinate with the Automated Requisition Tracking System (ARTS) to obtain the delivery information associated with each item being procured. When the item requisition is received by the purchasing department the purchasing agent will use the Automated RAMP Logistics Support System to communicate with suppliers. The Item Requisition will contain all technical information to buy or build the requested item. The purchasing agent will send a Request for Bid to the supplier using the Electronic Data Interchange (EDI)system and transmit the information in an ANSY standard. The supplier will respond with a Response to the Request for Bid using EDI and the associated ANSY standard. The purchasing agent will select the supplier and sent an electronic Purchase Order to the selected supplier.

7.1.4.1.5 Projected Item Delivery Messages

The response to the Item Requisition message from PP&C (Requisitioning) to RAMP FCIM will be the Projected Item Delivery message which is a query to the ARTS. The Projected Item Delivery messages provide RAMP FCIM with the purchasing information for items required to complete the order. If all items for the order have met the availability and delivery rules, then the order information will be filled out in the Projected Item Delivery message query of ARTS. If one of the approved substitute items, from the technical data information was selected for purchase, that information will also be reflected. If an item is not available PP&C (Requisitioning) will place the item requisition on hold and will send the Projected Item Delivery information to ARTS reflecting this problem. The RAMP FCIM will query ARTS, and upon receipt of the information reflecting this problem, RAMP FCIM will send an Order Status message to PP&C and place the order on hold.

RAMP FCIM will query ARTS to receive the Projected Item Delivery message(s) information identifying the new delivery date. If the item(s) required to fill an order do not meet the delivery requirements, then RAMP FCIM will then send an Order Status message to the PP&C notifying them of the new delivery date. PP&C will notify the customer of the new delivery date and will receive authorization to proceed or to cancel the order. PP&C will then send an Order Confirmation message to RAMP FCIM to cancel the customer order or to accept the customer order with the new delivery date. The RAMP FCIM system will send a Corrective Action Plan message to the ED&DB requesting a suitable substitute for the part. The ED&DB will coordinate with the Cognizant Technical Authority. The Cognizant Technical Authority will respond with a Design Exception Notice and ED&DB will send a Design Exception Notice message to RAMP FCIM providing an available suitable substitute. The ED&DB will use this information to update the technical data and forward the updated Technical Data Package reflecting the new part number to RAMP FCIM. RAMP FCIM will then send an Item Requisition message to the PP&C (Requisitioning) to order the available suitable substitute part.

If there is no suitable substitute available, the Cognizant Technical Authority will notify RAMP FCIM through ED&DB using a Cognizant Technical Authority Disposition message. The Cognizant Technical Authority Disposition message will state that the assembly must be redesigned to replace the part that is not available. RAMP FCIM will also send an Order Status message to the PP&C notifying them of the problem. PP&C will coordinate with ED&DB to obtain there design completion schedule. PP&C will also notify the customer of the problem and provide the customer with a scheduled completion date. If the customer does not accept the new delivery date, the customer may cancel the order. If this occurs, an Order Cancellation message canceling the order will be sent to RAMP FCIM from PP&C. This action requires RAMP FCIM and RASSP management intervention. The RAMP FCIM manager must call the PP&C (Requisitioning) and request cancellation of the purchase orders in process.

7.1.4.1.6 Request Quality Service Message

At RASSP the Quality Assurance functions will be within RAMP FCIM and will perform the day to day QA functions. For Quality Assurance requirements that are beyond the RAMP FCIM internal QA functions RAMP FCIM will coordinate with the RASSP Quality Assurance activities.

During the macro and micro process planning functions, if it is determined that external quality assurance functions are required, RAMP FCIM will send a Request Quality Service message to the RASSP Quality Assurance activity. This message will request that specific quality functions be performed and the Quality Assurance activity will respond with a Quality Service Commitment message to RAMP FCIM. RAMP FCIM will send a Part Quality Request message along with the parts to QA. When the functions are complete, QA will respond with the Quality Service Report message and return the parts to RAMP FCIM receiving.

7.1.4.1.7 Shipment Forecast Message

During the macro and micro process planning functions, the Shipment Forecast message is sent to the Packing and Shipping activity. This message provides the shipping materials information that will be required. If special packaging is required, this will give the Packing and Shipping activity time to obtain the materials.

7.1.4.1.8 Order Inquiry and Order Status Messages

PP&C may send an Order Inquiry message to RAMP FCIM requesting the status of a customer order at any time. RAMP FCIM will respond to PP&C with an Order Status message documenting where the customer order is in the process.

The RAMP FCIM system uses the Order Status message for communications between RAMP FCIM and PP&C. The following is a brief list of its uses:

- Reject Order
- Technical Data Reject
- Design Exception Notice
- Alternate Delivery Date

- Committed Delivery Date
- RAMP FCIM Order Completion
- Order Completion
- Order Status

7.1.4.1.9 Items To Be Shipped Message

During the processing of a customer order, RAMP FCIM will send an Items To Be Shipped message to the Packing and Shipping activity for items to be shipped out for external processes, for Quality Service, and/or when a customer order is complete.

When items are to be shipped out for an external process, for which the purchase order has already been generated, the shipping information will be included in the Items To Be Shipped message. When the work is completed and the items are returned, the RASSP Receiving activity will send an Item Receipt message to RAMP FCIM along with the items. This will notify RAMP FCIM of the return so that the customer order can continue its processing.

When a customer order has completed processing within RAMP FCIM, the Items To Be Shipped message will be sent to RASSP Packing and Shipping which will include all shipping instructions. RAMP FCIM shipping will package the items,along with all documentation, and forward the items to RASSP Packing and Shipping activity. RASSP Packing and Shipping activity will complete the shipping transaction and send a Shipment Information message to RAMP FCIM. The Shipment Information message will contain the final shipping information that will allow the customer order to be completed.

7.1.4.1.10 Shipment Information Message

Upon receipt of the Shipment Information message by RAMP FCIM, which signals to RAMP FCIM that a customer order is complete, RAMP FCIM will send a Parts Shipped message to Accounting.

7.1.4.1.11 Equipment and Operator Time Data Message

A message will be provided to RASSP, on a near real time basis, that supports an operator changing Project Control Numbers (PCN). This message will include direct and indirect PCN information. RASSP will query the RAMP FCIM Common Data Base for management information relating to Equipment and Operator Time Data on an as needed basis.

7.1.4.1.12 Corrective Action Plan and Quality Report Message

The RAMP FCIM system will request assistance from the Cognizant Technical Authority, through the ED&DB, during processing and test if a part cannot be fabricated/assembled in accordance with the technical data, or if the test results are not in accordance with the technical data. For these types of problems, RAMP FCIM will send a Corrective Action Plan message to the Cognizant Technical Authority via the ED&DB suggesting a corrective action, if known, ora description of the problem.

RAMP FCIM will also send a Quality Report message and an Internally Generated Technical Data message to the Cognizant Technical Authority, via the ED&DB, describing all action taken to the point where the problem was identified.

RAMP FCIM also sends a Corrective Action Plan message to the Cognizant Technical Authority, via the ED&DB, when obsolete components are identified fora customer order. This message identifies the obsolete component and requests a suitable substitute for the item. This may result in an Engineering Change Proposal (ECP) and/or Engineering Change Order (ECO), new Technical Data message from ED&DB, and/or a Cognizant Technical Authority Disposition message explaining what is to be done.

7.1.4.1.13 Cognizant Technical Authority Disposition Message

The Cognizant Technical Authority will use information in the Corrective Action Plan message, Quality Report message and/or the Design Exception Notice message to analyze the problem. If a design change is required, the Cognizant Technical Authority will return a Cognizant Technical Authority Disposition message to RAMP FCIM, via ED&DB, describing the product changes. If the problem can be corrected without a design change, the Cognizant Technical Authority will send a Cognizant Technical Authority Disposition message, via ED&DB, describing the change to the process and/or the change to the product that does not affect fit, form, function, or effectivity.

7.1.4.1.14 Preventive Maintenance Request Message

The RAMP FCIM system will send a Preventive Maintenance Request message to RASSP's Equipment Maintenance. This message will schedule Preventive Maintenance for RAMP FCIM hardware, software, and equipment and will be scheduled on a weekly, monthly, or quarterly basis. This schedule will be established by RASSP during RASSP integration.

7.1.4.1.15 Maintenance Schedule Message

Equipment Maintenance will send a Maintenance Schedule message to RAMP FCIM committing to the Preventive Maintenance requirements described in the Preventive Maintenance Request message. This schedule information will be stored in the database, and customer orders processed in RAMP FCIM will be scheduled around the Preventive Maintenance downtime, if scheduled during the processing shift.

7.1.4.1.16 Maintenance Outage Request Message

If a catastrophic failure occurs in the RAMP FCIM hardware, software, and/or equipment, RAMP FCIM will send a Maintenance Outage Request message to Equipment Maintenance for service. This message will describe the location of the failure and the type of problem. Equipment Maintenance will respond with a Maintenance Committed Time message. During this type of failure, RAMP FCIM will not be able to process work through the affected area and the maintenance departments will be required to provide a rapid response.

7.1.4.1.17 Maintenance Committed Time Message

RAMP FCIM will receive a Maintenance Committed Time message from Equipment Maintenance. Upon completion of the repair work, the Maintenance Committed Time message will be resent to RAMP FCIM; this time it will include information on repair, cost, labor, and labor time.

7.1.4.1.18 Current Interfaces in Use at RASSP

At present the interfaces to the existing FFM's are accomplished over the existing RASSP network electronically and physically. The interface messages support the islands of automation presently installed at RASSP. RAMP FCIM will integrate the FFM's under the RAMP FCIM architecture for total control. RAMP FCIM will also add controls for each FFM to communicate with the central ASRS and the AGV system.

7.1.4.1.19 Site Interface Capability in RAMP FCIM

To exchange information between the RASSP site enterprise and the RAMP FCIM implementation, an Interface Requirements Specification will be developed to support specific information (message) transmission.

7.1.4.2 System Level Components

There are two system architectures proposed for integration into the RASSP site enterprise: the RAMP Product Data Translation System (RPTS) is shown in Figure 7-10. The RAMP architecture consists of eight TLCs: 1) RAMP Control System, 2) Communications, 3) Information Management System, 4) Site Interface,5) Production and Inventory Control (P&IC), 6) Manufacturing, 7) Quality, and 8) Manufacturing Engineering. The RAMP FCIM architecture required to support a single printed wiring assembly factory floor system is shown in Figure 7-11.

Since the RAMP FCIM architecture supports modularity and flexibility Figure 7-12 provides a view of the same architecture supporting multiple factory floor modules. These modules are Printed Wiring Assembly, Printed Wiring Board Fabrication, Hybrid Micro Electronics assembly as a subset of the PWA, Cable Harness Assembly, Sheet Metal Fabrication, and Small Mechanical Part fabrication.

The RASSP implementation of each of these TLCs and the product data translation capability are discussed in following paragraphs.

7.1.5 Product Data Translation Issues

7.1.5.1 Product Data Translation Functional Requirements

In order to support the transfer of design information from the RASSP design system to the RAMP FCIM fabrication and assembly system and to convert existing designs for redesign within RASSP, the RASSP system is required to translate product data from human interpretable to a computer (CAD/CAE)interpretable format. The human interpretable product data currently resides on media such as aperture cards,

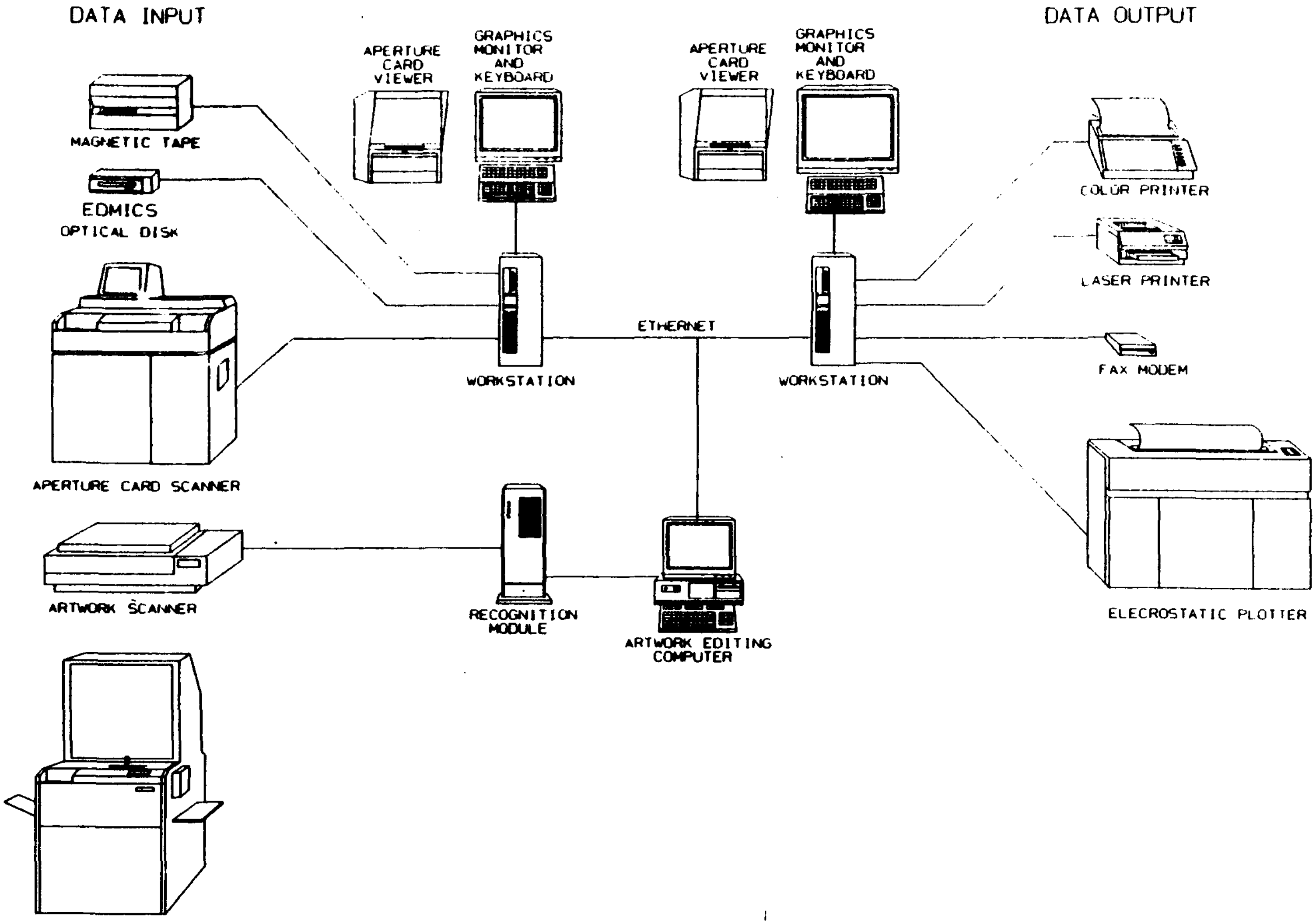


Figure 7-10. RAMP Product Data Translation System (RPTS).

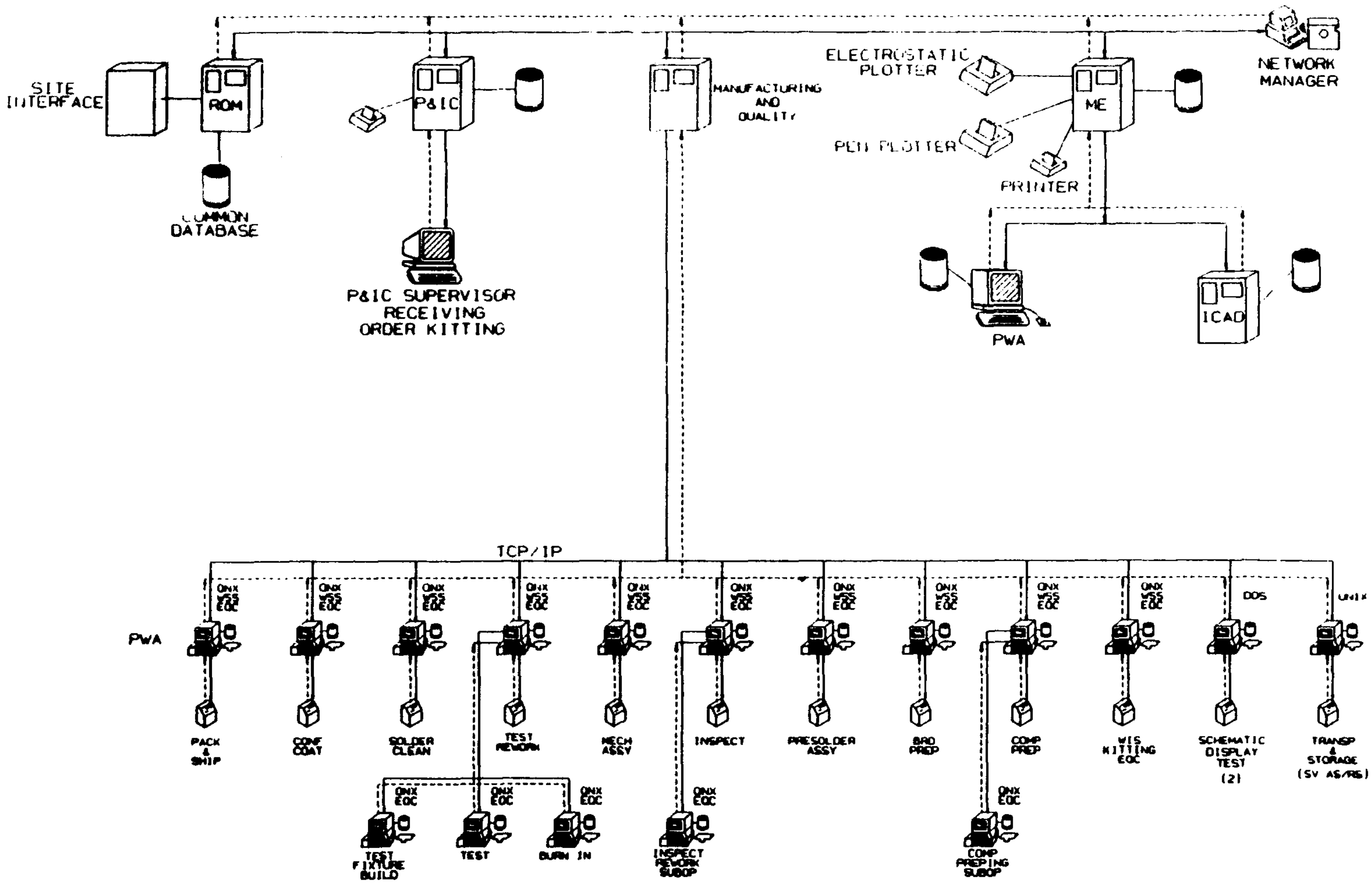


Figure 7-11. RASSP FCIM Architecture Supporting a Single PWA Factory Floor Module.

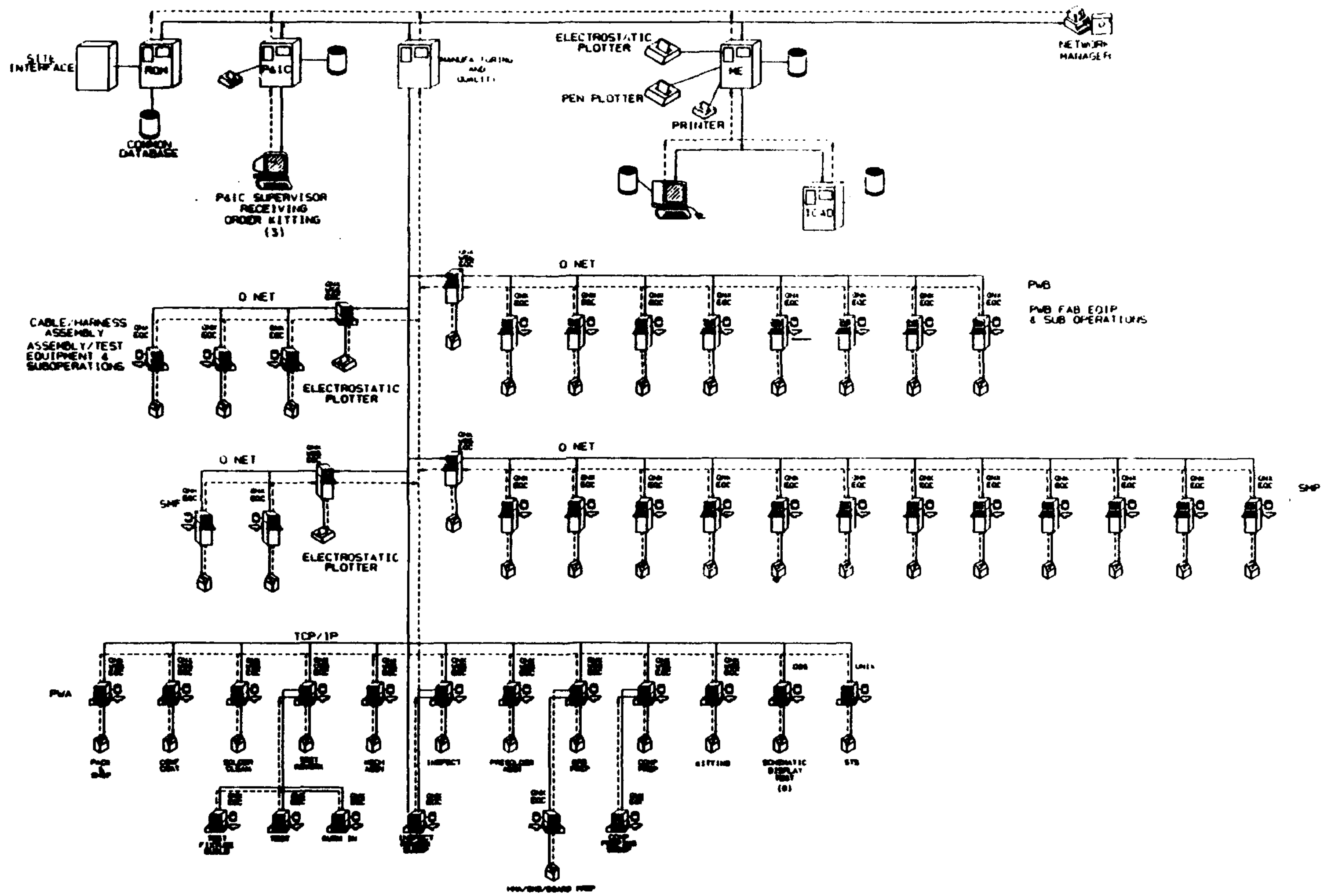


Figure 7-12. RASSP FCIM Architecture Supporting Multiple Factory Floor Modules.

microfilm, paper and raster digital data on various electronic storage media. This will allow the RASSP system to support there design, fabrication, and assembly of existing and new products.

The RASSP RPTS may be required to produce and accept product data in STEP or CALS compliant industry standard electronic files that can be exchanged by dissimilar CAD/CAE systems, and should be upgradable to support these and evolving CALS compliant standards that may be issued. The preliminary requirements for the RASSP RPTS are listed below.

- Product Data Types: Printed Wiring Board, Printed Wiring Assembly, Hvbrid Microelectronic assembly, Prismatic and Turned Machined Parts, Sheet Metal Parts, Cable Assemblies.

- Incoming Product Data Format and Media: Paper, Aperture Cards, Microfilm, CALS Type I Raster. Upgradable to include IGES, EDIF, STEP (mechanical products), Gerber and IPC-D-350, VHDL and STEP for electronic products. Bundled electronic product data files must comply with MIL-STD 1840A(or successor). Proprietary CAD files created on the Prime/CV CAD/CAE systems in RASSP. Media for electronic files must be consistent with existing RASSP CAD/CAE configuration and expandable to include optical disk, floppy disk, ethernet and TCP/IP channel of a baseband network.

- Product Data Translation System Output Format and Media: CALS Type I, Raster, STEP (machined and sheet metal products), Gerber, IPC-D-350 and Prime/CV CAD/CAE files. Upgradable to include VHDL, EDIF, IGES and STEP for electronic products. Bundled files will comply with MIL-STD 1840A(or successor).

7.1.5.2 <u>Product Data Translation</u>

Two product data translation systems have been developed under the RAMP Program. Both are applicable to the RASSP requirements, however neither is currently implemented on the proposed RASSP design system. However, the system scan except design information from any CAD design system and convert the information to any RAMP FCIM system. The functional capabilities of each are summarized below.

<u>RAMP Product Data Translation For Mechanical Parts</u>: The system is capable of translating human interpretable drawings, on paper or microfiche media, into PDES files that are capable of being exchanged between CAD/CAE systems. This system has been tested and deployed to DOD activities. It has been used to translate product data for machined parts (SMPs), and can be adapted to translate data for SM parts.

The time required to capture the product technical data into the CAD system and translate it averages four hours or less. The CAD/CAE software used is a feature-based commercial product that facilitates rapid capture of the product's physical features and dimensions.

A PDES translator has been designed for the CAD/CAE system under the RAMP Program. An IGES translator and a SM module have been developed by the CAD vendor, Parametric Technologies Corporation (PTC). A CV model of the machined parts can be created from data obtained from other PDES translation activities and used to develop manufacturing instructions. The RAMP manufacturing system's Manufacturing Engineering TLC has the capability to do this. A PDES to PTC translator has not been developed by either the RAMP Program or PTC.

RAMP Product Data For PWAs: This system is capable of capturing human interpretable product data on paper, aperture cards or raster into a robust CAD/CAE environment and then translating the product data into output files that can be exchanged between dissimilar CAD/CAE systems. The applications to date have included PWBs and PWAs. The system is now being expanded to include Hybrid Micro Electronic items and is now being deployed to DOD activities.

The system developed under the RAMP Program does not use the same CAD/CAE system that is suggested for the RASSP design but will exchange product data with the system employed by RASSP using EDIF, IGES, IPC-D-350 and CALS Type I raster files bundled using MIL-STD 1840A. It can be adapted to employ the CAD/CAE system used by RASSP, but there will be a significant development effort to do so.

The system can read the output files generated by the RASSP design system and convert the information for use in the RAMP FCIM fabrication and assembly systems. The average time to capture and translate a product data package is 24-48 system hours.

7.1.5.3 Additional Product Data Translation Capabilities Required

The current RPTS systems provide most of the functional capabilities that RASSP requires. They have been applied to a subset of singular machined parts and PWAs.

The following functionality will have to be translated to the CV CAD/CAE environment to achieve product data capture and translation for PWBs, PWAs and HMAs:

- Transfer the component capture database to CV.
- Change component capture function to generate a 2D cell information file.
- Change component capture function to generate a symbol file for Schedit.
- Write 2D cell create program for 2D component placement.
- Write an assembly generate program for CV's CADDS which includes techniques for component construction and a layering scheme.
- Develop a process for capture of drawing data including the process flow through CV applications, QA points, connectivity compare, error handling and file management.
- Port the RPTS Order Manager to CV

- Port/develop a translation to ISFs and 1840A binding (defer until required)
- Integrate the aperture card scanner with the CV environment
- Integrate the Gerber artwork capture system with the CV environment
- Develop schematic post processor (defer until required)
- Write new operators manuals and training materials.

The following is the status of the required file sets needed to drive the RASSP RAMP FCIM manufacturing and/or assembly process:

- SMP: Available from previous RAMP development.
- PWA/PWB: Available from previous RAMP development.
- HMA: Adaptable from previous RAMP development.
- CHA: Modification of RAMP PWA file set required.
- SM: Modification of RAMP SMP file set required.

7.1.6 Automated Manufacturing Technology – RAMP Based Approach

The RAMP FCIM automated manufacturing technology architecture supports modularity and flexibility. This architecture can be tailored to support any printed wiring assembly design requirements. This architecture can also be tailored to support multiple factory floor models. This architecture is designed to be integrated into any existing electronic design and manufacturing enterprise. To accomplish this RASSP integration the enterprise interface requirements have been established and can be tailored to meet specific manufacturing requirements.

7.1.6.1 RAMP Control System and Communications

The RAMP Control System, which is recommended for RASSP, provides the centralized overall process control that supports the system functionality. Each process in the RAMP system consists of multiple Commercial-Off-the-Shelf (COTS)and SCRA developed software applications that support specific individual tasks in the manufacturing process. Multiple processes can be sequenced together to provide for support of more complex manufacturing tasks. This architecture is highly configurable and provides for customization of individual RAMP systems to any selected site.

The RAMP Control System consists of three major components:

- RAMP Order Manager
- Application Control Interface
- Command Status Services

7.1.6.1.1 RAMP Order Manager

The RAMP Order Manager (ROM) is the component of RAMP Control System that processes all information requests to be executed within the RAMP system. The ROM software is executed upon startup of RAMP system. ROM control is invoked by message received data.

The ROM contains a list of all processes the RAMP system is capable of supporting. Contained in the list for each process is the order of applications that are required to support the process. ROM has the ability to alter the flow of control of the applications based on the completion code of the previous application. The ROM supports multiple requests for any single process, and a multiple number of active processes can exist in the system at any one time.

The ROM initiates a process when a Process Request message is received. Upon receipt of the Process Request message, the message is validated for content and the process table is searched for the existence of the requested process. After the requested process is found, the first application in the process's application list is executed. The ROM tracks each application's execution as it progresses through data download, application initiation, and data upload cycles. If a failure is detected, the ROM will stop the process and corrective action is taken. Once the problem is cleared, the process continues. After the last application in a process list is complete, the process is considered complete.

To support ROM design, integration, and test for the RASSP project requires integrating the process and application tables of the PWA and SMP. This integration also requires modification of the code that manipulates process and application names. In addition, new process and application tables for PWBs,HMAs, CHAs and SM will be developed and integrated into the ROM to support these capabilities as they are integrated into the RAMP system.

7.1.6.1.2 Application Control Interface

The Application Control Interface (ACI) software provides for the integration between the RAMP Control System TLC and the COTS packages used to support the operational functionality within the RAMP system. The ACI software determines which application command file is required for message support based on the message type received from the ROM. The application command file is specific for each application supported by the COTS software. The ACI software supports the interactive user in determining work to be performed, reporting the work as complete, and providing the users' availability to notify the ROM software that the user is ready to receive work. The ACI software also supports the ability to manually insert messages into the system when error conditions occur that require human intervention.

Work to be performed to tailor ACI for RASSP includes the generation of application command files that will be used to invoke both COTS and Developed Items (DI) software applications. There are also unique configuration files for each ACI that will be used in the RAMP system.

7.1.6.1.3 Command Status Services

The Command Status Services (CSS) software provides the message routing function and inter-process communication which are necessary to ensure that the Request/

Status and Command message types are sent to the desired TLCs that support the operations of the RAMP System. The CSS component also invokes the File Transfer Protocol (FTP) commands to send and/or receive the data files associated with the downloads and uploads to and from the common database.

The CSS component is started at system startup and continues to execute until a "stop" command is received, or is terminated at the command line via the system stop command. The CSS component supports the Inter Process Communication(IPC) function required for Control System components use. The CSS software provides a centralized and configurable point from which the IPC mechanism can be implemented in a heterogeneous system like the RAMP system.

A mailbox is used to pass messages to each item of software in the Control System TLC. Each software item will have a designated mailbox from which it will receive all messages, and a designated mailbox to which all out going messages will be sent.

All software items will write to the input mailbox of the CSS software. The CSS software will retrieve the message from its input mailbox and will determine the software item to which the message is to be routed to, based on the contents of the message and message type.
The CSS software also supports the transfer of the data files used in the downloads/uploads from/to the common database. The FTP function is invoked to make the actual transfer. The CSS software provides the necessary parameters to the FTP function that are required to make the transfer take place.

Work to be performed for RASSP will be in the area of the Mailbox configuration file. This file contains information on every RAMP message such as: source and destination of the message, FTP action necessary, and type and size of the message. This file is used by the ROM, CSS, and ACI components of the RAMP system.

7.1.6.2 Information Management System

FCIM systems rely on the design and population of databases that describe products, processes and resources employed in the manufacturing system.

7.1.6.2.1 RAMP FCIM Databases

The RAMP system uses relational databases that contain data about the processes used to make products, the factory resources, materials, job status,costs, product features, quality and similar information. The RAMP system database is distributed between the TLCs. Information that must be used by more than one TLC must be stored in the RAMP Common Data Base (CDB). This data base is resident on the main system computers and is accessible by the RAMP control system. Information that is used by only one TLC (such as ME reference DB) or is "in process" can be isolated to that TLC's use.

The CDBs implemented in the RAMP PWA and SMP include the following:

CAPACITY_PROBLEM	NDIRECT_ITEM
COMPONENT_BIN	ITEM_REQUISITION
COMPONENT_BIN_DPN_VIEW	ITEM_REQUISITION_VIEW
COMPONENT_BIN_ME_VIEW	ITEM_REQ_ME_DLD_VIEW
DEFECT_CODE	ITEM_REQ_ME_ULD_VIEW
DEFECT_PART_NOTICE	JOB
DEFECT_PART_NOTICE_VIEW	JOB_DPN_VIEW
ENG_SERVICE	JOB_MFG_VIEW
ENG_SERVICE_VIEW	JOB_OPERATION
FILE_CHECK	JOB_OPERATION_DLD_STAT_VIEW
FULL_DEFECT_VIEW	JOB_OPERATION_DPN_VIEW
JOB_OPERATION_VIEW	SITE_COG_TECH_AUTH
JOB_VIEW	SITE_COG_TECH_AUTH_VIEW
MATERIAL_LOCATION	SITE_CORR_ACTN_PLAN
OPERATION	SITE_CORR_ACTN_PLAN_DPN_VIEW
OPERATION_FILES	SITE_CORR_ACTN_PLAN_FORM_VIEW
OPERATION_FILES_DPN_VIEW	SITE_CORR_ACTN_PLAN_VIEW
OPERATION_ME_VIEW	SITE_DSGN_EXCPT_NOTICE
OPERATION_MFAC_VIEW	SITE_DSGN_EXCPT_NOTICE_VIEW
OPERATION_MFG_DPN_VIEW	SITE_DSGN_EXCPT_NOT_FORM_VIEW
OPERATION_MFG_VIEW	SITE_END_ITEM_SHIP
OPERATION_PIC_OP0_VIEW	SITE_END_ITEM_SHIP_VIEW
OPERATION_PIC_VIEW	SITE_EQPT_OPER_TIMEORDER_ACTION_OPTION
SITE_EXCESS_MATERIAL_FORM_VIEW	ORDER_PART_VIEW
SITE_EXT_SHIP_VIEW	ORDER_STATUS_VIEW
SITE_ITEMS_SHIPPED_FORM_VIEW	PART
SITE_MAINT_OUTAGE	PART_ORDER_MEIG_VIEW
SITE_MAINT_OUTAGE_VIEW	PART_ORDER_MEMP_VIEW
SITE_ORDER	PPIR
SITE_ORDER_DPN_VIEW	PPIR_JOB_ID_SEQ
SITE_ORDER_VIEW	PPIR_REQUEST_ID_SEQ
SITE_PARTS_SHIPPED_FORM_VIEW	QA_COG_TECH_AUTH_VIEW
QA_QUALITY_SERVICE_COMMITMENT	SITE_QUALITY_SERVICE_VIEW
QA_QUALITY_SERVICE_REPORT	SITE_SHIP_FORECAST_FORM_VIEW
QA_QUALITY_SERVICE_REQUEST	SITE_STATUS
RAMP_ORDER	SPACE_CHECK
RAMP_ORDER_DPN_VIEW	SUBMIT_JOB_1
RAMP_ORDER_VIEW	SWO_STATUS_VIEW
REQSEQ	TOTE_TYPE_QTY
SELECT_SCREEN_OPTIONS	SALGRADE

7.1.6.2.2 Additional Common Databases Required at RASSP

Several new CDBs will be required for the implementation of the RAMP architecture at RASSP. These additional databases will be required due to the incorporation of SM fabrication, PCB manufacturing, HMA and CHA into the RAMP architecture.

For example, a SM fabrication database must contain data on:

- Sheet Metal Cutting, Shearing, Sawing, and Braking Processes
- Sheet Metal Forming, Edge Forming, Rolling, and Bending Methods
- Sheet Metal Notching and Slotting Processes
- Sheet Metal Grinding and Deburring Methods
- Punch Types and Methods
- Welding Types and Methods

- Sheet Metal Fastener Types and Securing Methods (Rivets, Crimping, Riv-Nuts, Bolting, etc.)
- Plating and Painting Methods
- Engraving Methods
- Sheet Metal Drilling and Tapping Methods

A CDB for HMA must contain data on:

- Equipment/process specifications (range, limits, tolerances), actual usage, SPC data
- Overall fabrication trim and assembly rules and limits
- Certified consumable material specifications (resistor, conductor, dielectric)
- Certified consumable materials list and actual results SPC file
- HMA CNC file (Machine Programs)
- HMA Instructional Graphic File (pointers to archive)
- HMA STD Routing File (pointers to archive)
- HMA Test Program File (pointers to archive)
- HMA Digital Product Data (DPD) File
- HMA Process Time Standards history
- HMA Process Time actual (SPC) history data
- HM fabrication yield data
- HMA yield data
- HM/SMD Component specification data
- HMA/SMD Military and Commercial processes Standards data (if not included in PWA database)

7.1.6.3 Production and Inventory Control

This subsection describes how production and inventory control will be implemented in the RASSP FCIM system.

7.1.6.3.1 RAMP Production and Inventory Control Functions

The primary functions of Production and Inventory Control (P&IC) are:Order Entry, Material Inventory Management, Capacity Requirements Planning, Reserve Capacity and Production Control.

7.1.6.3.1.1 Order Entry

Order Entry has the following functions: Initiate Order, Confirm Order, Cancel Order and Determine Order Status.

7.1.6.3.1.1.1 Initiate Order

The purpose of Initiate Order is to receive RAMP order data and validate the data contents. Initiate Order will verify that part technical data exists for the order. The order is either accepted or rejected by a Production Supervisor.Initiate Order determines whether a PDES/STEP or a NON-PDES part is processed. Initiate Order indicates processing results to Determine Order Status.

7.1.6.3.1.1.2 Confirm Order

The purpose of Confirm Order is to allow the continuation of the order process following an exception processing situation. If a late material delivery date or a capacity problem delays production past a specified required delivery date, an alternate delivery date is suggested. Confirm Order either accepts the alternate delivery date or cancels the order. If a material substitution is suggested for a requisitioned material, Confirm Order either accepts or cancels the order.

7.1.6.3.1.1.3 Cancel Order

The purpose of Cancel Order is to allow the order to be canceled until the order is released to the shop floor.

7.1.6.3.1.1.4 Determine Order Status

The purpose of Determine Order Status is to examine and provide status, on a solicited basis, such as Order Inquiry, and an unsolicited basis such as Material Substitutions and Alternate Delivery Date situations. Determine Order Status will provide specified reports upon request.

7.1.6.3.1.2 Material Inventory Management

The purpose of Material Inventory Management is to manage the requisition of items required to produce the order, to receive a projected Item Delivery Date for each requisitioned item, to physically receive the item at RAMP and to determine when all requisitioned material is on hand.

7.1.6.3.1.2.1 Requisition Maintenance

The purpose of Requisition Maintenance is to receive both a Bill of Material(Item List) and Operational Routings from Manufacturing Engineering, and to create an item requisition with a purchase order number assigned to each item.Requisition Maintenance handles required items from Macro Process Planning, Micro Process Planning and Capacity Problem Planning. Requisition Maintenance determines the delivery date needed for the requisitioned item and informs Determine Order Status. Requisition Maintenance receives a projected item delivery date for each requisitioned item and then informs Capacity Required Planning when all dates are received. Requisition Maintenance manages material substitution and material required for rework situations. Requisition Maintenance manages indirect material as required by Manufacturing.

7.1.6.3.1.2.2 Item Receipt

The purpose of Item Receipt is to receive each item requisitioned, verify that an open requisition exists and that the quantity is correct. Item Receipt verifies that the requisitioned item is received at the site with all required data before it is received at RAMP. Item Receipt notifies Production Control when all requisitioned material is on hand.

7.1.6.3.1.3 Capacity Requirements Planning

The purpose of Capacity Requirements Planning (CRP) is to ensure that required delivery dates can be met by reserving and allocating a period of time for orders and Shop Work Orders (SWOs) to be processed at Manufacturing Engineering and Manufacturing resources. CRP ensures that the workload does not exceed the finite capacity available at these functions. CRP uses a forward scheduling approach to determine when jobs are to be performed, recognizing that a finite capacity constraint exists for all shop resources.

7.1.6.3.1.4 Reserve Capacity

The purpose of Reserve Capacity is to reserve a manufacturing start date for RAMP orders following the completion of Micro Process Planning and the receipt of all requisitioned items. Reserve Capacity reserves micro seats following the receipt of all projected item delivery dates.

Reserve Capacity also reserves enough capacity to produce the quantity specified in the order using a work station calendar and machine utilization factor to determine the first available reservation at each work station. Reserve Capacity determines an alternate delivery date when material delivery will delay production past the order-specified required delivery date and then notifies Determine Order Status. Reserve Capacity notifies Manufacturing Engineering that an alternate routing should be attempted if machine capacity does not allow manufacturing to be completed by the required delivery date.

Reserve Capacity processes data from Capacity Problem in the form of routings and requisitions. The Production Supervisor may extend the operating schedule if Manufacturing Engineering is unable to provide an alternate routing for parts with capacity problems. Reserve Capacity commits the delivery date if a capacity reservation is successful.

7.1.6.3.1.5 Production Control

The purpose of Production Control is to release SWOs when all items and process plans are available and shop capacity permits. Production Control creates SWOs, determines that all items are available, assigns priorities to SWOs,releases SWOs to Manufacturing, updates status as each SWO processing completes,and releases the order to the customer.

7.1.6.3.1.5.1 Create Shop Workorder

The purpose of Create SWO is to create SWOs for later release to the shop floor. Create SWO creates one SWO for each part in the order quantity. (Note P&IC This will be modified for batch (lot) orders in SMP and SM at RASSP.)

7.1.6.3.1.5.2 Release Shop Workorder

The purpose of Release SWO is to release the SWOs to manufacturing as prioritized by Assign SWO Priority and Required Delivery Date.

7.1.6.3.1.5.3 Assign Shop Workorder Priority

The purpose of Assign SWO Priority is to allow the Production Supervisor to set priorities on SWOs, put SWOs on a hold status and remove SWOs from a hold status.

7.1.6.3.1.5.4 Release to Customer

The purpose of Release to Customer is to verify the completion of the order,update order status to complete, notify Quality when part pedigree is required,and close out the order.

7.1.6.3.1.5.5 Update SWO Status

The purpose of Update SWO Status is to receive all completed operations from Manufacturing for each SWO and delete its work station reservation.

7.1.6.3.1.5.6 Indirect Inventory

The purpose of the Indirect Inventory system is to manage indirect items in the SYMIX system. Indirect items are automatically ordered through the requisition system either by low stock level or shelf life dates. Indirect item data is passed daily to the CDB for ME to match against during Macro Process Planning. All indirect items must be manually entered into the SYMIX system.Indirect items are issued to workstations using SYMIX transactions.

7.1.6.3.2 Current Production and Inventory Control requirements for RASSP

RASSP P&IC is interactively controlled using two standard systems: the enterprise MPR System and any Maintenance Shop Floor Control System (MSFCS). An Automatic Storage and Retrieval System and Automated Guided Vehicle system can be integrated with the MSFS system to form an effective P&IC system.

7.1.6.3.2.1 The SDS System

The SDS system is a collection of modules that provide data processing for the following functions:

- Work Measurement
- Cost Accounting/Budgeting
- Production Planning and Control -- Maintenance
- Production Planning and Control -- Supply Operations

The SDS is used at much higher levels of control than the RAMP system and does not have the detailed functionality at the lower level of manufacturing operations. SDS is not now used at the shop floor level of operations.

7.1.6.3.2.2 The MSFS

The MSFS is a set of computer software modules within the SDS. It is essentially an inventory control and tracking system. The MSFS interfaces directly with the RASSP ASRS and AGV system. The RAMP system will use the RASSP ASRS and AGV system for material storage and delivery to the RAMP Order Kitting function. Therefore, this will be one of the major interfaces between RAMP and MSFS.

There is a MSFS module used for time and attendance called Automated Time and Attendance Personnel System (ATAAPS). All data is entered by the shop floor supervisor for all personnel.

The RAMP system includes time tracking for all jobs at each operation and can be used to directly enter detailed time and attendance information into the ATAAPS. This will be another major interface between RAMP and MSFS.

7.1.6.3.3 Additional Production and Inventory Control Capacity Required for RASSP

This section describes additional RAMP functionality required to implement P&IC at RASSP.

7.1.6.3.3.1 Order Entry

The RAMP P&IC system receives orders from the site and releases them to ME for macro process planning after checking that data is available. When the macro process plan is finished, a workstation specific routing with time required at each station is put into the CDB. The P&IC system maintains a work calendar for all workstations that it uses to estimate delivery time and capacity problems. The RASSP RAMP will contain workstations for processes that have not been addressed by the RAMP system. These are: PWB, HMA, CHA, and SM. These workstation calendar models will have to be added to the workstation calendars that already exist for PWA and SMP.

The large quantities of SM and SMP parts, with their associated lot sizes,will require that the RAMP use batch processing. This basically means that only one work order will be generated for a group of identical parts, instead of one work order for each part as is done for PWAs. Tracking and Statistical Process Control (SPC) data will be by lot for orders handled in this way.

The P&IC system will expect to receive RAMP orders from the RASSP Program Control Branch. It is assumed that digital product data (DPD) exists for all subassemblies or fabricated parts that are expected to be produced.

If the RAMP does not receive separate customer orders for each subassembly or part, the ME ICAD system will request exception processing for each subassembly or fabricated part that it detects DPD file(s) exists for. It will do this by searching its data base of RAMP parts, similar to searching the indirect inventory data base. The ME will request that P&IC assign and enter a separate work order for each subassembly or fabricated part detected by the ICAD system. If the subassemblies also contain RAMP parts, the processes will repeat until all RAMP parts have been assigned separate work order numbers. When assemblies that do not have RAMP DPD for the assembly, but do not have it for fabricated parts (SMA) are detected, the system will also require exception processing.

7.1.6.3.3.2 Requisition Maintenance

At RASSP, Requisition Maintenance may process requisitions for assemblies as well as components. Requisitions may be for outside services, non-RAMP shops and RAMP shops which may require different processing for each situation.

Some BOM items may generate a new order to fabricate an item which will require tracking. A part need date is determined by Requisition Maintenance and clarification is needed as to what rules to follow. Each work station will have to be identified along with its processing rules and dependencies.

Data created by Requisition Maintenance will be passed to MSFS in a format to be determined. Requisition Maintenance may receive a "projected item delivery"in the form of a message rather than a date which indicates a problem.Requisition Maintenance may have all items placed on hold because one requisition has a problem which needs resolution, or one or more requisitions may need to be canceled, or the entire order may be canceled if the problem with the requisition cannot be resolved. Commercial part numbers may be input to "projected item delivery"in place of BOM part. Requisition Maintenance may trigger a Shipment Forecast message.

Item Receipt will verify material by comparing package labels to the requisition. Item Receipt will also handle testing of components and assemblies,when required, by interfacing with Manufacturing Engineering. A barcode tag will be placed on each receipt.

Item Receipt interfaces with MSFS in a format to be determined. MSFS interfaces with the ASRS for storage of the item. When all items for the order have been received, Item Receipt requests all items to be sent to the kitting workstation. Item Receipt also processes indirect items and interfaces with the MSFS and ASRS. Item Receipt receives the "site receipt" for an external process.An "items to be shipped" will be sent to the Quality Department along with the item so required tests can be performed. The item will then be received again with the quality documentation as part of the "site receipt" record.

7.1.6.3.3.3 Production Control

Production Control creates the necessary SWOs to support the order quantity. The current system creates one SWO for each part of the order quantity. SWOs may need to be created based on a lot size to support batching. An alternate method is to group

SWOs with a quantity of 1 together to form the "batch". Releasing the SWOs to the shop floor requires an interface to ASRS in order to retrieve the parts for the order.

7.1.6.3.3.4 Capacity Requirements Planning

Capacity Reservations will occur for both probes and firm orders. The RAMP system design effort will determine what rules to follow for probes versus firm orders.

7.1.6.3.3.5 Indirect Inventory

The current PWA Indirect Inventory is managed by SYMIX. Storage is in a shelf type environment where the parts are taken in and out of storage by an operator using SYMIX transactions. The RASSP ASRS is where the indirect inventory will be stored. There needs to be an interface between the RAMP Indirect Inventory and both the ASRS (for storage and retrieval) and the MSFS (for procurement.)

If RAMP is responsible for the automatic ordering by low levels and shelf life expiration date, it is imperative that the on hand quantities be accurate.RAMP must know when inventory is increased or decreased through the ASRS. An interface must be maintained that keeps RAMP informed whenever on hand quantities change. If shelf-life materials are to be handled, it is also imperative that RAMP know the location and shelf life expiration date for each shelf life material that RAMP is responsible for.

7.1.6.4 Manufacturing Engineering

Manufacturing Engineering functions are those that relate product data,process data and manufacturing resources to develop routing, fabrication,assembly, inspection and test instructions for use on the shop floor.

7.1.6.4.1 Current Manufacturing Engineering Capabilities at RASSP

RASSP has a large and robust Prime/CV CAD/CAE system which they now use to develop NC programs. They do not use CAD/CAE for process planning.

7.1.6.4.2 Manufacturing Engineering Capabilities in RAMP

The function of the Manufacturing Engineering TLC is to produce the process plans used to manufacture parts in the RAMP facility. A process plan includes the workstation routing sequence and instructions for operations to be performed at each workstation during the manufacture of the part. The instructions include all machine programs, operator instructions, and graphics needed for the manufacture of the part.

The process plan generated by Manufacturing Engineering includes any operations required to complete the process that must be performed outside of the RAMP system. In addition to the outside processing operation, the routing also includes the shipping operation which precedes the outside service, and the receiving operation following the outside service. The process plan includes any acceptance test instructions to be executed after the part has been manufactured.

The Manufacturing Engineering process supports different types of process planning methodologies. These methodologies include variant, hybrid and generative planning techniques.

In addition to creating and modifying process plans, Manufacturing Engineering also provides information to the P&IC, Manufacturing, and Quality processes. Manufacturing Engineering supports P&IC by providing process time estimates; determining stock requirements; and generating requests for all items required to manufacture the ordered part including tools, assembly fixtures, test equipment, and test fixtures. Manufacturing Engineering supports the Manufacturing process by providing Engineering Services. Manufacturing Engineering supports the Quality process by providing problem evaluation services and recommending corrective actions for quarantined parts.

The following is a brief description of the operations performed by the Manufacturing Engineering process. (Please refer to RAMP Document SCR004003-0 for a more complete description of the functionality.)

7.1.6.4.2.1 Create Process Plan

This process is responsible for all functions related to the creation or modification of process plans at the RAMP facility. These functions include, conversion of the Technical Data Package from a STEP or CALS-compliant format to RAMP-specific format; creation and refinement of the actual process plans through either variant, generative, or hybrid planning techniques; creation of the final test and/or inspection plan; and, maintenance of databases used in process planning.

7.1.6.4.2.2 Evaluate Problem Cause

This function is responsible for supporting the manufacturing process by providing Problem Cause and Corrective Action Plan output to Coordinate Disposition of Quarantined Part. An analysis is performed to determine what caused the problem, i.e., was the Process Plan executed correctly and was the shop process operating within control limits. From this analysis, decisions are made regarding the processes that were/should be used to manufacture the part and the disposition of the part.

This function is also responsible for resolving improvements to process efficiency. Process problems that may not be part related, are identified and investigated by Manufacturing Engineering to determine their cause and develop corrective actions.

7.1.6.4.2.3 Differences Between SMP and PWA Manufacturing Engineering

The SMP Manufacturing Engineering (ME) system is configured to receive ISF,PDES and proprietary CV files. Data from paper can be manually entered by the operators. When PDES technical data for parts is transmitted to SMP RAMP, along with an order, ME can utilize the generative Macro Process Planning Function. The Generative Process Planning Function will analyze the PDES files, develop Item Requisitions, and develop a Macro Process Plan.

The committee for PDES standards has not released the approved specification for feature based PWA PDES files. The PWA RAMP system is presently configured to receive ISF or proprietary CV technical information, and as a result is limited in its use of the Generative Process Planning Functions. Technical data from paper can be entered by the operators. Therefore, the Generative Process Planning Function is augmented with a robust Variant Process Planning and Browser capability. This allows the manufacturing engineer to edit existing process plans to achieve the desired results.

7.1.6.4.3 Additional Manufacturing Engineering Capabilities Required for RASSP

The current RAMP manufacturing engineering capabilities for PWA and SMP are directly useful in the RASSP system. Manufacturing engineering in support of SM parts has much in common with the RAMP SMP. Manufacturing engineering for the HMAs has much in common with the RAMP PWA, with some variations. The DPD fileset for PWBs is already in use in the RAMP PWA, but the manufacturing engineering capabilities to fabricate the PWB must be developed. The CHA operation is similar to the RAMP PWA Mechanical Assembly workstation. Manufacturing engineering can be customized to the cable task, making it more automatic than Mechanical Assembly, especially for downloading test programs to the automated cable test systems.

7.1.6.4.3.1 Additional Digital Product Data File Set Development Required

The following is the status of the required DPD file sets to drive the manufacturing process:

SMP:	Available from previous RAMP development or CV files.
PWA/PWB:	Available from previous RAMP development or CV files.
HMA:	Adaptable from previous RAMP development or CV files.
CHA:	Modification of RAMP PWA file set required or CV files.
SM:	Modification of RAMP SMP file set required or CV files.

7.1.6.4.3.2 Additional Macro Process Planning Capabilities Required

The two main activities in macro process planning are generating the material requisition file and creating a route the work pieces follow among the resources within the FCIM cell. The existing system features which generate the material requisition file will be extended in a straightforward manner to include all planned categories of work. The overall file structure will provide sufficient classification information to assign a given file to one of the six work categories. The following paragraphs describe the additional automatic routing capabilities needed for the four new categories.

7.1.6.4.3.2.1 Additional Automatic Routing Capability - HMAs

The artificial intelligence engine within macro process planning will route the substrate through the required processes, based on the generic types of materials to be applied, such as: conductors, insulators, dielectrics,resistors, etc. In like manner, it will then route the complete substrate through the attachment and connection of wire bond connected integrated circuit chips and surface mounted devices, based on their generic device types. A study of allowable manufacturing sequences will be made to establish the necessary inspection and test steps in the process. Provision will be made to ask for human assistance if the rule base cannot reach a decision for an unusual set of circumstances.

7.1.6.4.3.2.2 Additional Automatic Routing Capability - CHA

The artificial intelligence engine will choose which operations within the CHA workstation are required, based on the generic type of cable (integral,built-up, etc.) and the generic type of connector terminals.

7.1.6.4.3.2.3 Additional Automatic Routing Capability--PWB Fabrication

The basic distinction to be made in routing PWBs is between single-sided,double-sided, multi-layer, and flexible PWBs. As many needed decisions about the detailed processes as can be made, based on the incoming DPD file set, will also be forwarded for use in micro-process planning.

7.1.6.4.3.2.4 Additional Automatic Routing Capability--SM Fabrication

The fundamental techniques used to develop the automatic routing for SMPs are applicable to SM fabrication, except the processes are different. A new set of artificial intelligence rules will be adapted from SMP to fill this need.

7.1.6.4.3.3 Additional Process Planning Capabilities Required

The micro process planning system generates a process plan for each sub-operation included in the route for a product being manufactured. This plan includes a script file, which commands the human and machine activities,numerical control (N/C) files which drive various kinds of automatic equipment,and instructions and graphics for human operators.

Any additional categories Required to support the selected architecture can be added to the system as identified. The identified information will be established from the RASSP design system rules for fabrication and assembly.

7.1.6.4.4 Manufacturing Engineering Implementation

Implementation of Manufacturing Engineering to support the RASSP FFMs will proceed in phases to support Phase One, Two and Three of the project. The RAMP PWA ME capabilities, with added ME data bases, and integrated COTS CAD and CAE applications to support component test, HP3070 ATE test, test fixture fabrication, automated conformal coat, and automated temporary solder mask application will support the Phase One system.

The Phase Two implementation will require addition of ME data bases as well as further integration of COTS CAD and CAE application packages. To meet the increased load of expanding the PWA throughput and adding the HMA, CHA, and PWB capabilities more workstation positions will be required.

Phase Three will add the basic RAMP SMP Process Planning capabilities for SMPs and adapt the ME processes such that they can also be used for SM process plans. This will require addition of specialized COTS packages to handle modular tooling utilizing pallet-handling horizontal milling equipment and CNC turning equipment that RASSP now has or will upgrade to in the near future. Postprocessors now utilized for CNC code generation will be integrated with the RAMP SMP system standard post processors which will give RASSP a very wide range of process planning capabilities.

7.1.6.5 Quality

This section describes the product and process quality functions to be implemented in the RASSP FCIM system.

7.1.6.5.1 Quality Functions in RAMP System

The Quality functions that support the requirements for the RAMP system are performed in the Manufacturing TLC, ME TLC, Site Interface TLC, and the Quality TLC. The Manufacturing TLC provides inspection data, part pedigree information,SPC charts, and discrepancy information. The ME TLC provides part routing,exception processing information, and inspection instructions. The Site Interface TLC handles the messages between the RAMP and the Site that support the Quality TLC. The Quality TLC collects quality information for reports, provides notification to the Quality Engineer (QE) of activities requiring QE response,and provides the QE with the capability to review Quality data.

Specifically, the Quality TLC: generates quality reports, coordinates the dispositioning of discrepant/quarantined parts, arranges for quality services not found within RAMP, assembles Part Pedigree Reports, generates part quality records, organizes and plots data for SPC activities, and monitors resource certification.

- The Generate Quality Reports function accesses and retrieves part quality data and inspection results, and processes them into reports which denote both the acceptability and the actual performance level of the manufacturing processes.

- The Coordinate Disposition of Quarantined Part function oversees the disposition action for the affected parts in Discrepant Part Notifications, tracking the evaluation, preparing corrective action plans, coordinating with external quality functions, and scrap decisions.

- The Assemble Part Pedigree Reports function assembles and compiles a complete component/material pedigree for a requested part.

- The Generate Part Quality Record function creates a historical file which includes all customer requested quality reports, quality reports required by internal policy, and Part Pedigree Reports.

- The Statistical Process Control function accesses and retrieves part quality data and inspection results, and processes them into Pareto Charts, Run Charts and Control Charts to support internal SPC activities.

- Monitor Resource Certification monitors and validates equipment certification, equipment calibration, and personnel certification processes within the RAMP.

The PWA RAMP implementation is designed for operation in compliance with the following military specifications:

- MIL-STD-1686
- Electrostatic Discharge Control Program for Protection of Electrical and Electronic Parts, Assemblies and Equipment
- MIL-I-45208
- MIL-Q-9858
- MIL-STD-2000
- MIL-STD-45662

7.1.6.5.2 Current Quality Processes for the RASSP System

The Quality Control and Quality Assurance programs presently in use at the RAMP FCIM system are governed by the regulations, plans, and instructions listed in Table 7-2. These documents provide the Quality Assurance requirements that a system is to comply with. Additional Quality Control and Quality Assurance requirements will be derived from the RASSP design rules and added to the RAMP FCIM system.

Table 7-2. Current quality plans and instructions

Regulation Number	Date of Issue	Title
AR 70-37	27 JUN 1991	Research and Development
DESCOM-R 702-1	20 SEP 1989	DESCOM Product Assurance Program
DESCOM-R 702-1-C1	6 DEC 1990	DESCOM Product Assurance Program
DESCOM-R 702-1-C2	13 MAY 1990	DESCOM Product Assurance Program
702-5	2 FEB 1984	Product Assurance for Preproduction/First Article Inspections
740-7	17 JAN 1990	Storage and Supply Activities for Electrostatic Discharge Sensitive/Fragile Item Control Program
750-15 CH-1	30 JAN 1990	Maintenance of Supplies and Equipment
750-15 CH-2	19 AUG 1991	Maintenance of Supplies and Equipment
Plan Number	**Date of Issue**	**Title**
702-144	DEC 1987	Quality Assurance Plan for Inspection of Electrostatic Discharge Sensitive Items
702-199	MAY 1989	Quality Assurance Plan Outlining the General Quality Assurance Provisions for Special Fabrication Projects
Instruction Number	**Date of Issue**	**Title**
SD-003	3 JUL 1986	Handling Electrostatic Discharge (ESD) Sensitive for Electronic Components

7.1.6.5.3 Additional Quality Capability Required for RASSP

In addition to the Quality Control and Quality Assurance requirement capabilities outlined in the previous paragraph, all Military standards covering PWAs, SMPs, PWB fabrication, HMAs, CHAs, SM fabrication, RPTS (internal Technical Data generation), and Component Inspection within the PWA/Microelectronics will be included. An example of the additional requirements that are not available in RAMP, at this time, are MIL-STD-1772 for microelectronics. This standard and all associated standards will be included in the development and integration of the HMAs system that will be integrated into the PWA system.

7.1.6.5.4 Quality Implementation

The required development within the RAMP architecture to accept the Quality Control functions, information, and data for the PWA and SMP systems are already designed into the present systems.

Development will be required to include the Quality Control functions,information, and data for: PWB fabrication, HMAs, CHAs, SM fabrication, RPTS(internal Technical Data generation), and component inspection within the RASSP RAMP system.

The activities required to develop the new Quality Control functions for RASSP include the following:

- Identify requirements that are not currently handled in the PWA or SMP, or a previously developed RAMP RASSP FFM
- Access the impact, if any, of the new requirements on the existing base system
- Identify redesign of existing system to accommodate the new requirements, if necessary
- Generate the required design documentation
- Design, code and test new and modified code.

8. RASSP APPLICATIONS

8.1 Top Down System Analysis For RASSP

We utilized a top down analysis to determine the military and commercial classes of systems benefiting most from the RASSP process. Many platforms in military campaigns include functions that require signal processing to complete a successful mission. Likewise many commercial systems require signal processing to perform a certain function. As a first step to select the classes of systems for RASSP, it is critical to determine the classes of systems important to the users and the customers.

The metrics or basis we used to determine the best classes of systems for RASSP can be described in two parts:

1. The relative importance of the signal processing function to a military mission was determined from the mission analysis and commonly used commercial systems that require signal processing were identified. Critical functions for military campaigns and commercial uses were considered good candidates for RASSP.

2. Once the important military and commercial systems were identified, present signal processor costs (acquisition and Life Cycle), time to specify, time to produce, time to field, and maturation potential data was collected for those important systems. Processors with high costs, long development times, and poor maturation potential are considered excellent candidates for RASSP.

Once a class of systems is identified for RASSP, the RASSP process will dramatically reduce processor development time, costs, and provide the capability for growth without incurring further costs. In order for the RASSP process to succeed, the initial requirements set by the customer must be used to rapidly define the design parameters (waveforms) for the analog and digital front end, and the processor. The present design process to define the front end waveforms based on the mission or commercial system requirements are performed without much automation or integrated tools. This results in design times that do not meet the short RASSP specification schedule requirement. Our approach is to automate, as much as possible, the process to design the parameters for the analog and digital front end and the processor to reduce the specification design time and cost.

Target Systems For RASSP

Military Systems: Our initial study had selected Automatic Target Recognition (ATR), Air-to-Air and Air-to-Ground Radar, ESM, and ECM as the best systems for RASSP. We will continue to study other potential classes of systems for RASSP. The selection was based on a top down analysis where the target systems for RASSP flows from and is traceable to the mission timelines generated from a Far East and Mid East campaign scenario. Using this approach, events in the timelines were analyzed and applied to the derivation of top level system requirements. The derivation continued into generation of signal processing requirements.

Cost, development time and maturation potential data were collected and used to determine the best target systems for RASSP. Figure 8-1 depicts the top down approach used to identify target systems for RASSP.

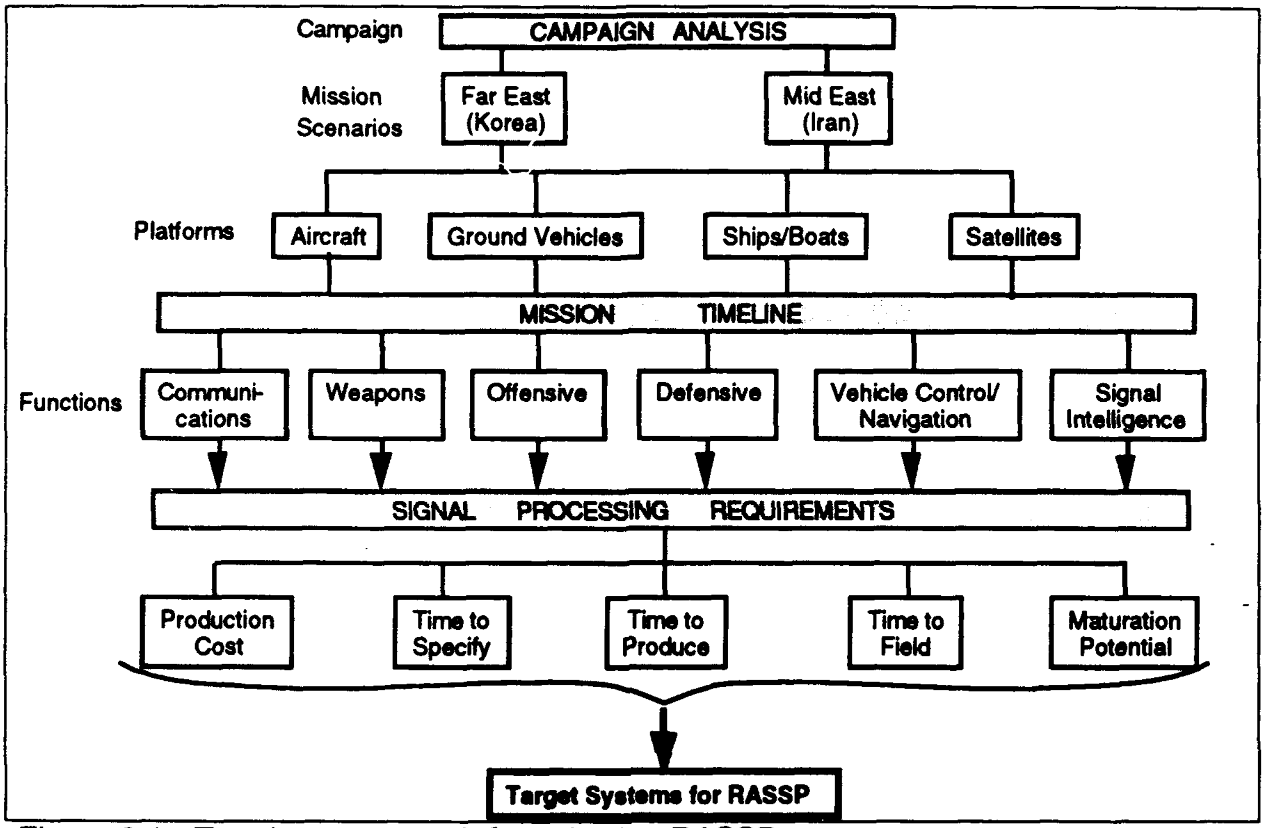


Figure 8-1. Top down approach for selecting RASSP target systems.

We used the SUPPRESSOR and ALARM mission models to run a simulated Mid East and Far East mission. The models integrated the effects of sensors, tactics, command and control structures, countermeasures, weapons, targets, and prescribed threat laydowns. In the simulation we considered aircraft, ground vehicles, ships, and satellites which were inputs to derive the event timeline. Simulated event timelines gave the required functions for each event during the mission. An example of a mission timeline for the Mid East Scenario is provided in Figure 8-2. This timeline exhibits only the major events for the mission. We expanded the timeline to include the detailed events of the mission. That is each minute of the missi.n was analyzed and required functions were identified for the significant events.

The importance and frequency of certain functions are made apparent through the timeline event analysis. Those functions that require signal processing can be traced back to the mission events providing a means in which to define the front end signal processing waveform requirements.

The results of the mission timeline analysis were consolidated into a top level matrix as a summary (see Table 8-1). The platforms considered in the campaign analysis are listed in the first column. Potential classes of systems required for the mission are listed across the top of the matrix. The x's indicate the platforms that require the

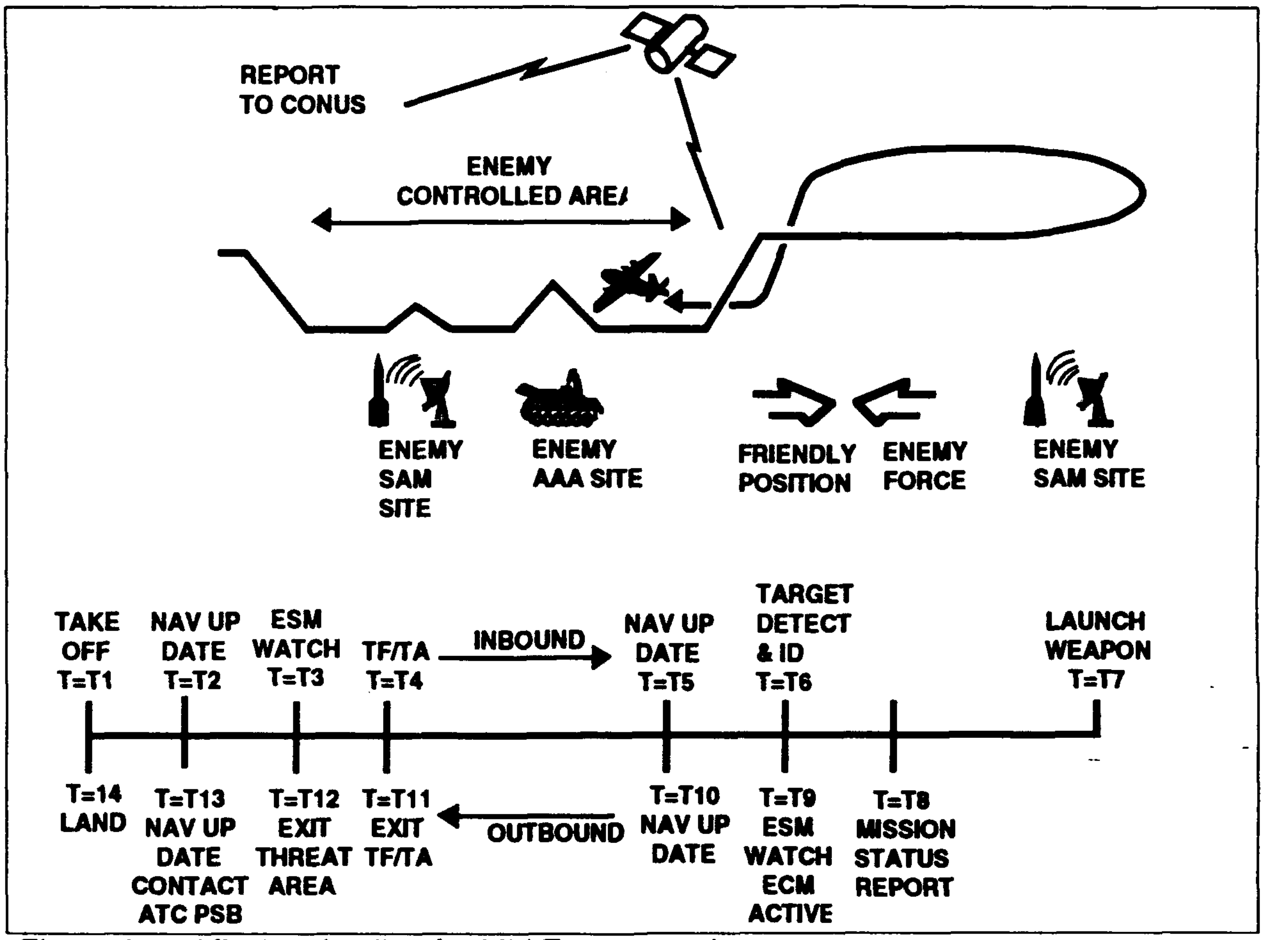


Figure 8-2. Mission timeline for Mid East scenario.

various classes of systems for the mission. The hi-lighted columns indicate those classes of systems most used and most important to the mission. For example ATR is considered essential for a high probability of kill and survival for Navy, Army, and Air Force vehicles in the Far East scenario. This scenario has a large threat density in a high priority target area resulting in a single pass multiple target requirement. The abnormally high workload for the operator(s) in this area requires some functions to be automated, specifically target recognition. ATR allows the operator to concentrate on the defensive systems and operation of the vehicle while the targets are automatically detected, identified, and classified. With the operator able to concentrate more on his survivability, the probability of mission success increases.

ATR is performed through sensor data fusion algorithms. Various sensors, such as Radar, IR, ESM, and Communications provide range, polarization, cross section, azimuth, thermal structure, frequency, and other waveform parameters (as depicted in Table 8-2). The input data is processed to give a high statistically probability of target recognition. These algorithms are process and memory intensive for a real time application. Presently processors are becoming available that have the CPU power to meet these severe ATR processing requirement. As more powerful processors mature into the field ATR will be a fundamental function in many military platforms.

Table 8-1. Classes of systems for RASSP.

PLATFORMS	RADAR	Data/ Comm	ECM	ESM	ATR	FLIR	IRST	NAV	SONR	INTEL	ID	TF/TA	LSR RDR	LASER DETECT	WTHR RADAR
ARMY															
Tanks		X				X							X		
Artillery															
Attack Helo	X	X	X	X	X	X		X			X				X
A-10	X	X	X	X		X		X			X				X
NAVY															
AX	X	X	X	X	X	X		X			X	X		X	X
F-14	X	X	X	X	X		X	X			X				X
F-18	X	X	X	X	X	X	X	X			X				X
F-5	X	X	X	X				X			X				X
A-6	X	X	X	X		X		X			X				X
Subs		X	X	X	X			X	X		X				
Battleships	X	X		X				X			X				X
Destroyers	X	X						X							X
Carriers	X	X	X	X				X							X
AirForce															
F-22	X	X	X	X	X	X	X	X			X	X		X	X
F-15E	X	X	X	X	X	X	X	X			X			X	X
F-111	X	X	X	X	X	X		X			X	X			X
F-117		X	X	X	X	X		X			X				
F-16	X	X	X	X	X	X		X			X				X
B-1B	X	X	X	X	X			X			X	X			X
B-2	X	X	X	X	X	X		X			X				X
F-4	X	X	X	X	X			X			X				X
AWACS	X	X	X	X				X		X	X				X
C-130	X	X	X			X		X			X				X
Satellites	X	X				X				X					X
WEAPONS															
Cruise Missile	X	X			X			X							
Hellfire Missile		X				X		X							
AAMRAM	X							X							
AIM-9						X		X							
GBU-16		X						X					X		
GBU-24		X						X					X		

Table 8-2. Typical sensor waveform values for ATR.

PARAMETER	RADAR	FLIR	ESM
Wavelength	N/A	8-12 μm	N/A
Frequency Range	7-11 GHz	N/A	0.5 to 18 GHz
FOV	N/A	1.0°	N/A
Power (Avg.)	125W	100W	250W
Duty Cycle	40%	N/A	30%
Instantaneous Bandwidth	2.0 GHz	N/A	2 GHz
Sensitivity	-70 dBm	0.08 K	-85 dBm
Dynamic Range	60 dB	25 dB	70 dB
Azimuth Accuracy	± 5°	N/A	± 10°
Throughput Delay	10 ns	5 ns	6-12 ns
Input Data Rate	600 Mbits/sec	380 Mbits/sec	200 Mbits/sec
Gain	30 dB	N/A	35 dB
Pixel Size	N/A	0.04mm x 0.04mm	N/A
Pulse Width	6 μsec	N/A	>100 ns
Pulse Density (max)	4,650,000 pps	N/A	100,000 pps
Input Impedence	50 ohms	300 ohms	50 ohms
Resolution	10 ft	0.1 mrad	10 MHz
Frequency Accuracy	± 2 to 3 MHz	N/A	± 2 to 3 MHz pulse-to-pulse
Reliability	25,000 hrs	25,000 hrs	25,000 hrs
Weight	TBD	TBD	TBD
Frame Rate	N/A	60 Hz	N/A
Number of Pixels	N/A	100 (10 x 10)	N/A

Front-end and CPU hardware data was collected for those target systems identified as most important to the campaign shown in Table 8-1. This data included present off-the-shelf cost, time to specify, time to produce, time to field, and maturation potential for the analog and digital front-end and the processor. As shown in Table 8-3 the Radar, ECM, ESM, and ATR signal processors are the most costly, have long development times, and have no maturation potential, thus very good candidates for the RASSP process.

The data collected in Table 8-3 is based on hardware supplier data for current off-the-shelf signal processors for the various functions. We weighted cost, specification time, production time, fielding time, and maturation potential equally in determining the best candidate systems for RASSP. We will continue to work with DARPA and the services to refine and modify the list of candidate systems for RASSP. We will also continue to update our supplier data base for off-the-shelf signal processor hardware and software cost, development times, and growth potential.

Table 8-3. Current processor development data.

TARGET SYSTEMS	Production Cost ($)	Time to Specify (mos)	Time to Produce (mos)	Time to Field (mos)	Maturation Potential
Military					
A-A/A-G Radar	95,000	4-5	8-9	6	none
ECM	50,000	3	8	6	none
ESM	65,000	6	8	6	none
ATR	125,000	6-8	8-10	6-8	none
Weather Radar	15,000	4	6	5	none
Communications	15,000	3	5	4	none
UHF/VHF Radios					
HF Radios					
SATCOM					
JTIDS					
JSTARS					
Navigation	20,000	5-6	7-8	3-4	none
GPS					
MILSTAR					
TACAN					
ILS					
MLS					
Identification	12,000	3-4	4-5	3-4	none
IFF					
Beacon					

Commercial Systems: We analyzed the commercial market for potential candidates for the RASSP process. Our top down approach studied five major categories of systems including aircraft, ground vehicles, ships/boats, commercial satellites, and consumer electronics. A data base was developed for specific products in each of those categories, that require signal processing, and performance requirements for each product was defined. Figure 8-3 illustrates the approach for selecting candidate commercial systems for RASSP.

Commercial and private aircraft, automobiles, trucks, trains, commercial ships and boats, consumer alarm systems, High Density TV, and law enforcement radar were

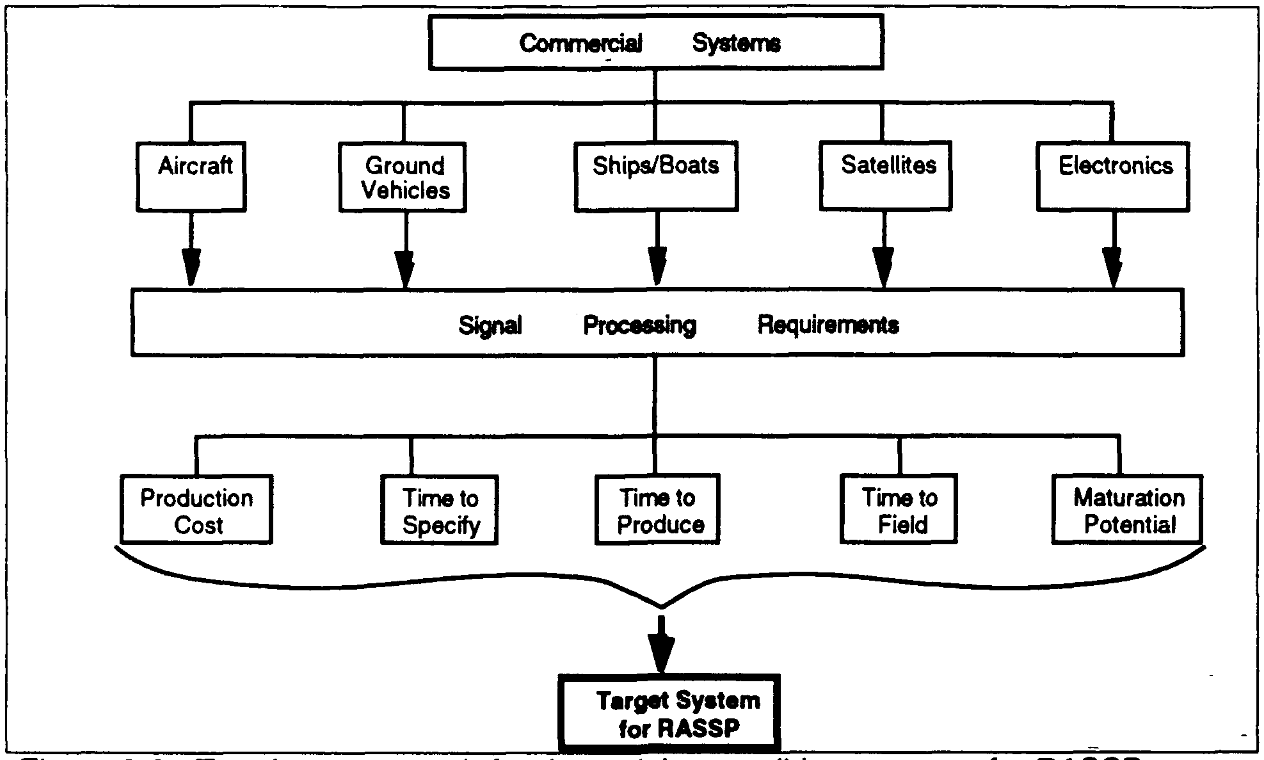


Figure 8-3. Top down approach for determining candidate systems for RASSP.

the products and systems considered for RASSP. These products or systems have several important functions that require signal processing. The functions are:

- Commercial/Private Aircraft
  - UHF Radio
  - TCAS
  - Microwave Landing System (MLS)
  - Weather Radar
- Autos, Trucks, Trains
  - Commercial/Private Ships and Boats
  - Radio
  - Radar (including Weather)
  - GPS
- Law Enforcement, Alarm Systems, HDTV
  - RF
  - Infrared
  - Video

We recognize Weather Radar and Microwave Landing Systems for commercial aircraft are growth items. New commercial aircraft such as the Boeing 777 and the MD-12 will likely require those functions. Weather Radar and MLS will provide added safety for the passengers and crew as well as making the flight smoother.

For each function we collected cost, time to specify, time to produce, time to field, and growth potential data for the analog and digital front end, and the processor. This data base is summarized in Table 8-4.

Table 8-4. Commercial systems selected for RASSP.

System	Signal Proc. Function	Cost ($)	Time to Specify (mo.)	Time to Produce (mo.)	Time to Field (mo.)	Growth Required?
AIRCRAFT						
Commercial	UHF Radio	10,000	0	2	3	no
Private	TCAS	25,000	4-5	7-8	3-4	possible
	*Microwave Landing Syst.	50,000	5-7	6-8	6-8	possible
	*Weather Radar	40,000	6	8	7-8	yes
GROUND VEHICLES						
Automobiles	*GPS	5,000	6	6	5-6	yes
Trucks						
Trains						
SHIPS & BOATS						
Commercial	Radio	1,000	0	2	1	no
Private	Radar (incl. weather)	8,000	2	3	2	yes
	GPS	5,000	0	2	1	possible
COMMERCIAL ELECTRONICS						
Law Enforcement	Radar	1,500	0	1	1	no
Alarm Systems	Infrared	200-5,000	0	1	0	possible
HDTV	Video	3,500	4-5	6	2-3	yes

* Growth Items

Radar for commercial aircraft and ships and boats was identified as good a candidate for RASSP due to its relatively high costs and long development times including the analog and digital front end and processors. Similarly MLS was also considered a good candidate for RASSP. The airlines, to remain competitive, will require all new systems to be low cost and meet scheduled delivery dates. Most commercial ships and boats can only afford low cost radar systems. The RASSP process can significantly help to keep costs low and development times short. Low cost weather radar and MLS are also required for many military operations. Since there are military and commercial users that require low cost, short development times with maturation potential for the weather radar and MLS functions RASSP is an excellent process for the development of their respective processors.

GPS, a widely used system for precise navigation, is rapidly gaining popularity with commercial ships and private speed and sail boats. However, the GPS system must be affordable and readily available for large numbers to be purchased. The front ends and processor are major cost and schedule components of GPS. Reducing the processor costs and development times will help to facilitate an affordable and available GPS.

8.2 RASSP Application Demonstration Candidates

GE recommends that a proof of concept demonstration be performed on the program to benchmark the RASSP benefits. The demonstration should be a follow on to a well documented/existing design to enable a valid comparison of the RASSP implementation to the "then" implementation. Ideally the design should benefit from the Model Year concept (modular and extendable architecture, implementable with commercial technology, and realizable with open system hardware and software approaches). In addition, the application must demonstrate other key capabilities of RASSP: rapid prototyping, design for test, and design for manufacturability.

The purpose of performing the application is to benchmark the results achieved relative to performance, cost, and schedule effects, for all aspects of the design. Key benchmarks for monitoring performance are:

- Hardware design effectiveness
- Software productivity
- Virtual prototyping environment effectiveness versus actual build
- Automated manufacturing cost and cycle times
- Long term effects - life cycle cost improvements

The RASSP system provides varying degrees of benefits to a wide range of applications. Broad classes of applications relevant to RASSP, and their relative degree of payoff from RASSP are highlighted in Figure 8-4.

Category	RASSP Benefit	Processor/System Characteristics	Examples	Benefit
1	Lower NRE/Production Cost	High Volume Applications Low Cost Systems	• Expendables—Automatic Target Recognition	High
2	Retargeting of Designs for a Variety of Form Factors	Common Processor Functions Performed on a Range of Platforms—Air, Ground, Space	• Image/Data Compression • Communications Functions • Command/Control Functions	
3	Retargeting Common Designs to a Wide Range of Applications	Programmable Flexible Processor Architecture	• Common Airborne Radar, IR, Surveillance Processor	
4	Reduced Life Cycle Costs	Large Established Platforms which are Regularly Upgraded	• Shipboard (AEGIS) Systems • Submarine (BSY-2) Systems • Airborne (JSTARS) Systems	
5	Fast Prototyping	Leading Edge, State-of-the-Art, and One-Of-A-Kind Systems	• Spaceborne Experiments • DemVal Designs	
6	Integrated Electrical, Mechanical, Thermal Design Environment	Platforms With Severe Size and Environmental Constraints	• Selected Ground Vehicle & Space Platforms	Low

Figure 8-4. RASSP application payoff.

An initial list of potential system applications for RASSP demonstrations is provided in Table 8-5. Based on guidance and support from DARPA and other RASSP government offices, the GE team will brief the key RASSP concepts to pertinent program managers to determine the highest payoff applications, and to solicit support for the demonstrations.

Table 8-5. Potential system applications.

System Candidate	Point of Contacts	Benefit due to RASSP
Airborne Surveillance	(Class of Systems)	Airborne Surveillance systems are a critical component of modern defenses. Sophisticated signal processing and data correlation algorithms are required to improve performance. The use of embedded super-computers such as the Intel Touchstone products is currently being studied.
E-2C (APS-145)	Primary: Cmdr. Jim Maurer System Prgm Mgr: Capt. Sheperd Prime Contractor: Grumman/GE	The E-2C AEW radar has served the fleet for over 30 years. Designed to operate over water, upgrades in processing is required to operate near and over land and to detect low RCS targets such as cruise missiles and TBM's. New techniques such as STAP and sensor fusion need to be developed and demonstrated. Hence the capability to rapidly prototype processing systems is a critical need.
JSTARS	Primary: System Prgm Mgr Prime Contractor: Grumman/Norden	The JSTARS system proved it's value in the Gulf War. It is a new system and hence will require a series of upgrades as tactical experience uncovers new requirements and new technologies/algorithms are developed.
IRST	(Class of Systems) Prime Contractor: GE	IRST systems have recently become practical due to new developments in high throughput processing technology. As systems are deployed and used, new techniques to control false alarms, provide robust tracks, and fuse IRST information with other pre-existng sensors are required.
F-22 AIRST	Primary: Govt. Prgm Mgr: LTC. D. Wright Govt. COTR: Mr. R. Haren Prime Contractor: TBD (GE or Martin) GE Prgm Mgr.: G. McElroy COTR: Brian O'Toole (GE)	The F-22 AIRST is planned to be the first P3I to the baseline avionics system. A comprehensive Dev/Dem program is currently being procured. One of the key risks to be retired is the capability to provide sufficient processing power for look-down capability in the allotted volume and power. Algorithms to detect targets against clutter will be developed and tested over the next five years. The capability to rapidly develop prototype processors will be required to allow rapid turnaround of new algorithms as data collection proceeds.
F-14D IRSTS P3I	Prime Contractor: GE Government Contact: M. Sokloff (NADC) S. Campana (NADC) GE Contact: D. Acuri K. Fuhr	The IRSTS is the first deployed system on a US aircraft. The feasibility of adding look down capability is being explored. The primary equipment mod is the processor. The capability to rapidly prototype this processor, prove capability in flight tests, and insert the upgrade early during the production run could result in significant savings.
JIAWG Compatible Modules	Prime Contractor: Hughes	The JIAWG standard will drive the development of all new avionic processors. JIAWG has defined a standard interface between sensor front ends and processing resources similar to the RASSP goal. VHDL and GLSS are used to accelerate new module development. The concept of "model year" upgrades of CIP compatible modules would have a significant impact on LCC of new and existing aircraft.

RECOMMENDED PROGRAM PLAN

The GE team has completed the RASSP Phase I study, and has demonstrated the state of the industry and development trends in electronic design CAD, and in signal processor design, requirements and methodology. GE has assembled a world class team to address the development of the required RASSP advanced methodologies and design system. The team members have demonstrated their commitment to RASSP through participation in the study phase with minimal and in most cases no contract funding.

The following recommended plan is based on the identified requirements for a RASSP system, a recognition of development trends and anticipated industry accomplishments independent of RASSP, and judgment by the GE team of the areas where the government investment through the RASSP program can be most effective.

Recommended Approaches and Strategies

A large aerospace firm, experienced in all aspects of system design, signal processor design, and the RASSP requirements should lead the RASSP Phase II program; this will ensure proper focus on meeting the system assigned requirements are met, and will provide a rich set of demonstrations. The RASSP development team should include one or more larger aerospace firms that will serve as sites for porting the RASSP design system during the four year period.

The applications, and tool development strategies should be chosen to align with the DoD/Aerospace needs, and the EDA vendors interest (from a marketability perspective).

The Phase II program should build on existing approaches in an evolutionary manner — to build and utilize large integrated CAD tool systems, and to leverage lessons learned to the maximum extent. Systems with proven cost and schedule reduction capability provide the best base capability for RASSP to use as a starting point.

University/research organization developments should be utilized where creativity is required, however the associated risk needs to be effectively managed. Historically many of the EDA vendor concepts and tools have been initially developed at universities like CMU, Berkely and Stanford.

Large commercial EDA vendors should be used in their respective areas of expertise wherever possible, because of their inherent need to maintain their franchise and leverage large investment in their particular functional areas.

Existing standards will be utilized for the model year concept to the maximum extent feasible, to avoid both the long delays in establishing of new standards, and the risk of change associated with evolving standards. The RASSP Phase II program needs to establish and maintain involvement with DoD standards organizations, to influence developments and maintain currency with the trends of the organizations (example: JIAWG).

Multiple DoD applications should be identified that ca benefit from the RASSP system, and participation early in the program should be enlisted. Knowledge based aids, including synthesis tools and design advisors, should be an integral aspect of the RASSP design system supporting all tasks from conceptual design through interfacing to automated agile manufacturing facilities.

Coordination with ongoing DARPA/Tri-Service programs should be initiated and maintained, where significant mutual benefit can be realized (ex: MHDL, AHDL, PAP-E PIEE, etc.).

Linkage with the ASEM program developments needs to happen early in the program for improved efficiency and time to market with the MCM Technology.

RASSP virtual prototyping thrust should be coordinated with DoD Synthetic Environment Thrust 6 to eliminate duplication and obtain program leverage.

Task Overview

The tasks recommended (WBS format) for implementation for the RASSP program are identified below. The associated schedule for performance of the tasks is provided.

1. Methodology/Requirements Definition

   - Model Year Refinement
   - Process Model Development
   - Concurrent Engineering Methodology
   - Simulation Methodology
   - Signal Processor Interoperability/Scalability

2. Enterprise Infrastructure Development

   - Core Architecture Selection Process
   - Develop/Adopt Interframework Communication/Integration Approach
   - Data Representation for RASSP Design Objects
   - Definition/Implementation of RASSP Database Management Approach
   - Develop Integrated Simulation Backplane
   - Develop/Implement Synthetic Environment Approach
   - Enterprise Framework Integration
   - Enterprise Data Management and Control System (DMCS)
     — The DMCS functions that should be addressed in the definition of the approach are:
       - Process Management
       - Rules
       - Object Management
       - Configuration Management
       - Product Structures for Bills of Material
       - Product Structures
       - Object Navigation and Query
       - Revision and Version Processing

- Notifications
- Approvals
- Access control
- Check-In and Check-Out
- Trigger Task Execution
- Implementation and Customization

3. Electronic Integration and Commerce

- Enterprise Framework Electronic Integration with Manufacturing Centers
- Electronic Integration with RASSP Team Members
- Integration with Vendors Manufacturing Centers
- Integration of Enterprise Procurement Group with Suppliers

4. HDL/SDL Development

- HDL Extensions
- HDL Extension to Address Analog Design
- HDL Extension to Address Physical/Process Descriptions
- Definition/Selection of Compatible Software Description Language

5. System Tools/Analysis

- Definition of Top Level Design System and Tool Requirements
- Extension/Upgrade to Current Tools for a Unified Design/Simulation Environment
- Development of Seamless Interface Approach to Lower Level Design Tools
- Develop Integration Links to Design Advisors/Cost Estimation/Reliability Analysis Tools
- Extend Selected Data Flow Graph Based Tool Set to Support Multiprocessing
- Develop/Extend Automated Partitioning and Mapping Tools to Support RASSP Requirements
- System Level Synthesis Tools
- Develop Hierarchical Linkages Between System Analysis Tools Behavioral, Functional, and Data Flow Simulators
- Cost Estimation Model Integration
- RAM Model and Tool Integrations

6. Software Tools

- Autocode Generation Tool Extensions
- CASE Tool Development/Extension
- Signal Processing Algorithm Libraries
- µKernal Support Software

7. Design Advisor Development

- Testability Advisors/Testability Synthesis Capability
- Develop System Level Partitioning Advisor/Mapping Advisors

- Develop HDL Based Architectural Synthesis Capability
- System Level Tradeoff/Codesign Advisors
- Analog Design Advisors
- Design Advisor Manager Development:

8. Test Approach/Tools

- Develop Hierarchical DFT Test Strategy and Tool Support for Top Down Design Methodology
- Automatic Test Generation Tools
- Virtual Test Environment Support

9. Hardware Development Tools

A limited number of hardware development tools are recommended for funding under the RASSP development, in that it is anticipated that the program will leverage current technologies and the substantial ongoing developments in this area.

- Analog Hardware Tool Development/Extensions
- Mixed Mode Simulation Development/Extension
- Synthesis Tool Development/Extensions
- Detailed/Structural Design Tools

10. Automated Manufacturing

- Extension of FCIM Approach to Support PWB and MCM Enclosures
- Commercialization Support for RASSP Automated Manufacturing
- Manufacturing Integration with Design System for Synthetic Environment Support:
- Automated Test Generation
- Test Feedback for Engineering Enhancement

11. Library/Model Development

- Develop Integrated Component Information System Concept:
- Development/Extension of Component Libraries in Standard Format
- Develop Library Verification Methodology/Support Tool Set
- Algorithm Library Development
- Implement Model Generation Tools/Support

12. Industry Dissemination

- Installation of RASSP Systems at Team Member Sites
- Develop Industry Distribution/Support Center for RASSP
- Develop RASSP System Briefings and Training Courses
- Identification and Support of Contractor, Vendor, and DoD Beta Sites
- Establish Vendor Alliances for Support of the Model Year Concept

13. Application Development

- Joint (DoD/GE Team) Definition of Applications for RASSP Demonstration
- Implement High Impact Demonstrations
- Virtual Prototyping Capability
- RASSP System Benchmarks
- Application Demonstrations

14. RASSP System Integration

- Tool Set/Framework Integrations:
- Simulation Integrations
- Database Integration
- Demonstration Support

15. Program Management

Program Plan: Develop a program plan for execution of the program. Develop plan revisions as directed by the government during execution of the program.

Program Reviews: Conduct program reviews on a semi-annual basis, at either the contractor site.

Subcontract Management: Perform task associated with management of RASSP subcontractors including: completing procurement process in issuing subcontracts, monitoring of technical and financial progress, and providing technical and programmatic direction to ensure that the objectives of the subcontract are achieved.

Reports: Prepare and deliver to the government monthly and annual reports. Reports to address technical and financial status of the prime contractor and subcontractors.

CDRL List: Develop and produce required documentation on the RASSP design methodology and the design system. Anticipated CDRL items include training manuals, tool documentation, and process flow documentation for the various RASSP design processes, demonstration system and all aspects of the enterprise infrastructure and specifics of the framework.

Schedule

The following figure illustrates the recommended phasing of the tasks for the Phase II RASSP program. Highlights of the proposed program ares as follows:

System Demonstrations to be performed in each year:

Year 1 - Prototype CAD system with a limited application demo.
Year 2 - Alpha CAD system with Design Phase aspects of Application Demonstrated.

Year 3 - Beta CAD system with Advanced Design Phase and Implementation Phases of Application demonstrated.

Year 4 - CAD system with Manufacturing Center, and Enterprise Data systems integrated. Advanced CAD tool capabilities demonstrated.

RASSP Methodology and CAD system requirements defined in year 1.

RASSP Enterprise Infrastructure and Framework requirements derived from CFI requirements specifications.

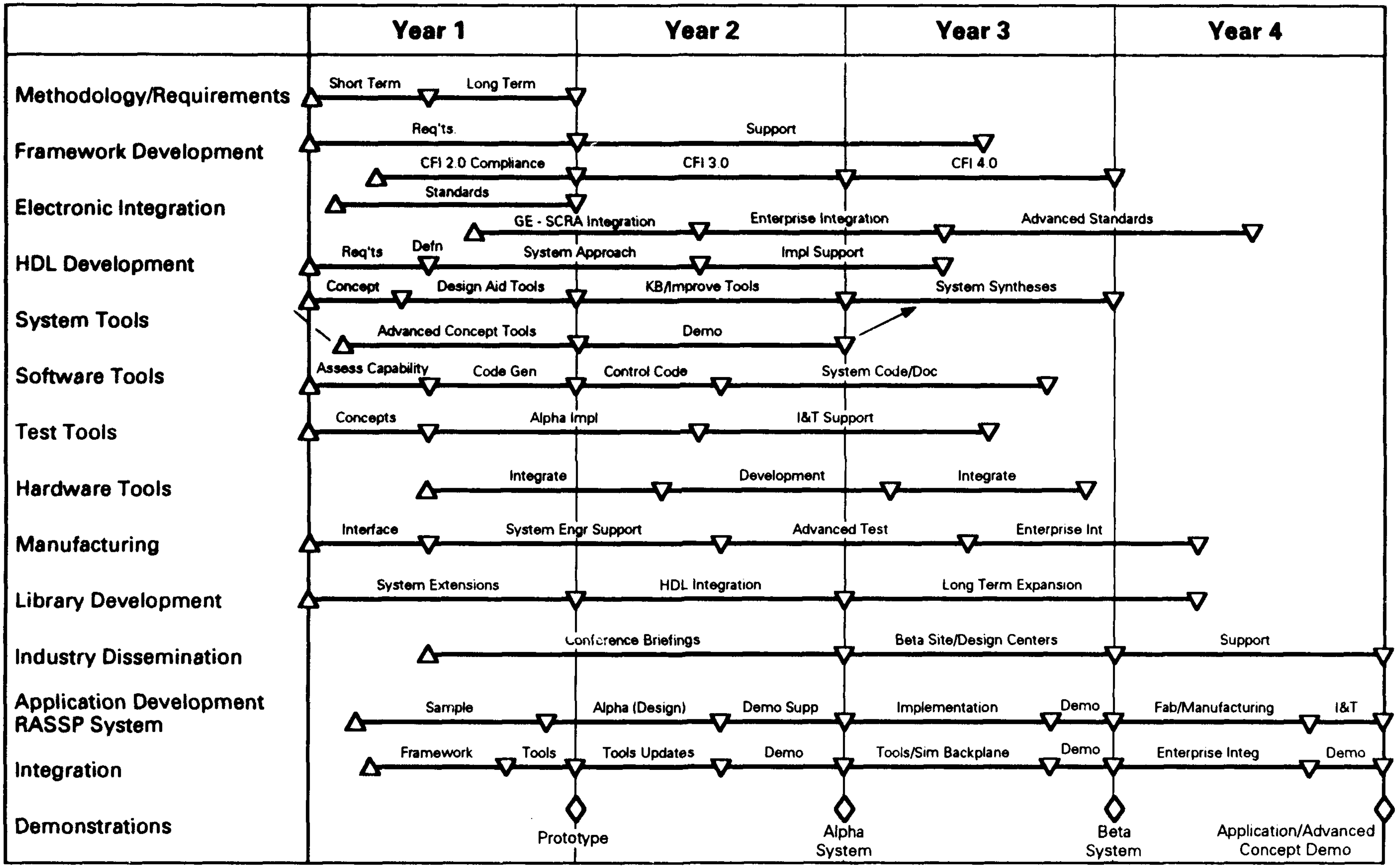
RASSP Framework developed via extensions to an existing/supported framework.

Multiple integration levels—short term and long term.

Multiple implementation approaches supported for higher risk tool developments—ex: system synthesis tools.

Implementation approach for lower risk tools.

Language development task as core activity; language integration/support by tools as part of tool integration tasks.



RASSP implementation program.