AD-A241 698

(2)

ANNUAL REPORT

VOLUME 7

TASK 7: VLSI DEVELOPMENT

REPORT NO. AR-0142-91-002

September 24, 1991

GUIDANCE, NAVIGATION AND CONTROL

DIGITAL EMULATION TECHNOLOGY LABORATORY

Contract No. DASG60-89-C-0142

Sponsored By

The United States Army Strategic Defense Command

COMPUTER ENGINEERING RESEARCH LABORATORY

Georgia Institute of Technology

Atlanta, Georgia 30322-0540

Contract Data Requirements List Item A005

Period Covered: FY 91

Type Report: Annual

91-12578

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No 0704-0188

1a REPORT SECURITY CLASSIFICATION Unclassified	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY 2b DECLASSIFICATION/DOWNGRADING SCHEDULE	3 DISTRIBUTION/AVAILABILITY OF REPORT 1) Approved for public release; distribution is unlimited 2) continued on reverse side
4 PERFORMING ORGANIZATION REPORT NUMBER(S) AR-0142-91-002	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION School of Electrical Eng. Georgia Tech 6b OFFICE SYMBOL (If applicable)	7a NAME OF MONITORING ORGANIZATION U.S. Army Strategic Defense Command
6c ADDRESS (City, State, and ZIP Code) Atlanta, Georgia 30332	7b ADDRESS (City, State, and ZIP Code) P.O. Box 1500 Huntsville, AL 35807-3801
8a NAME OF FUNDING/SPONSORING ORGANIZATION 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER DASG60-89-C-0142
8c ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO / WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)
Guidance, Navigation and Control Digital Emulation Technology Laboratory
Volume 7 (Unclassified)

12 PERSONAL AUTHOR(S)
C. O. Alford, W. S. Tan, S. H. russ, J.I. Chamdani, A Register, S. Gieseking, T. Kubota,

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Annual	FROM 9/28/90 TO 9/27/91	9/27/91	38

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

1. Introduction
   1.1 History
   1.2 Objectives
   1.3 REquirements
2. Existing GN&C VLSI Chip Set
   2.1 Executive Processor
   2.2 Data Processor
   2.3 Interconnection Network
   2.4 Signal Processor
3. Next Generation of Integrated Circuits
   3.1 Introduction
   3.2 Executive Processor
   3.3 Interconnection Network
   3.4 Signal Processor
4. Summary
5. References

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION Unclassified
22a NAME OF RESPONSIBLE INDIVIDUAL	22b TELEPHONE (Include Area Code) / 22c OFFICE SYMBOL

DD Form 1473, JUN 86 Previous editions are obsolete SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

Distribution statement continued

2) This material may be reproduced by or for the U.S. Government pursuant to the copy license under the clause at DFARS252.227-7013, October 1988.

ANNUAL REPORT

VOLUME 7

TASK 7: VLSI DEVELOPMENT

September 24, 1991

Authors

W. S. Tan, S. H. Russ, J. I. Chamdani, A. Register, S. Gieseking, T. Kubota, T. C. Huang, D. Pai, J. Pendegrass

COMPUTER ENGINEERING RESEARCH LABORATORY

Georgia Institute of Technology

Atlanta, Georgia 30322–0540

Eugene L. Sanders	**Cecil O. Alford**
USASDC	**Georgia Tech**
Contract Monitor	**Project Director**

DISCLAIMER

The views, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other official documentation.

DISTRIBUTION CONTROL

(1) This material is approved for public release unlimited distribution.

(2) This material may be reproduced by or for the U.S. Government pursuant to the copyright license under the clause at DFARS 252.227–7013, October 1988.

TABLE OF CONTENTS

1. Introduction

Under the Strategic Defense Command, KEW Directorate, Georgia Tech is developing a set of modular VLSI chips that can be used to construct a light weight, low power, and high performance flight computer to guide, navigate, and control (GN&C) advanced kinetic energy weapon (KEW) interceptors. This effort involves an in depth study of GN&C algorithms, modularization of the algorithms, implementation of the algorithms in VLSI chips, testing and evaluation of the chip sets as a system.

1.1. History

From 1975 to 1984, the Computer Engineering Research Laboratory at Georgia Tech was under contract with the Ballistic Missile Defense Advance Technology Center to develop advanced high performance computer architectures that are capable of simulating high fidelity control systems in real–time. The result of this effort was the discovery of a functional processing technology that enables the construction of parallel computers that can meet the stringent real–time processing requirements of high performance, complex control systems.

Since 1984, this technology has been applied to the design of a testbed that can be used to verify the functionality of flight hardware on the ground. The same technology is also being applied in the design of a set of VLSI chips for an on–board flight computer, except the chip designs will be converted into radiation hardened process.

1.2. Objectives

The primary objective of the GN&C research effort is to develop the technology necessary to construct a light weight, low power, high performance flight computer for guidance, navigation, and control of advanced KEW interceptors. The mission of the flight computer is to guide the interceptor to a point in space during the boost phase, receive update information and orient the interceptor to a designated target space during midcourse, track the targets and perform necessary maneuvers and divert operations to guide the interceptor into an incoming RV (reentry vehicle) at the terminal phase.

The bulk of the processing power is required in the terminal phase. During this phase, the flight computer must process images from a 128x128 focal plane array (FPA), perform various types of filtering operations on the images, and convert the images into object clusters for tracking.

1.3. Requirements

The basic required interfaces for the GN&C processor are to the Inertial Measurement Unit and the valves that control the various thrusters in the interceptors. This basic interface requires relatively low communication bandwidth with the GN&C processor.

During midcourse, it may be necessary for the GN&C processor to receive target information and orientation commands from the ground based (or space based) Battle Management Control Center. As a result, an interface from the GN&C processor to some type of telemetry link is required. This interface also does not require high data bandwidth.

The interface that requires the most bandwidth is the focal plane array (FPA). The size of the target FPA is 128x128 pixels. The processing rate for the images from the FPA is 100 frames per second. At this rate, the GN&C processor must perform all necessary filtering operations to separate the targets from the background noise. These filtering operations include non–uniformity compensation, temporal filtering, spatial filtering, and thresholding. After these various signal processing, the pixels are grouped into objects (clustering operation) and their centroids are calculated (centroiding operation). Once the targets are clustered, Kalman filtering is performed to track the movement and to extract the velocity of the targets. Discrimination techniques separate the targets from decoys. One of them is designated for the purpose of computing the final aim point. All necessary processing is then performed to guide the interceptor to the designated target.

Computations for the tracking and discrimination, as well as control processing, are carried out in IEEE, 32–bit, floating point numbers.

Based on the computing requirements for the guidance, navigation, and control of KEW interceptors, the GN&C processor is functionally decomposed into three general classes of processor architectures: executive processor (GT–EP), data processor (GT–DP), and signal processor (GT–SP), and executive processor (GT–EP). A fully–connected 8–point crossbar switch connects the various processor modules in a closely coupled interconnection network. Figure 1 shows the various functional modules of the GN&C processor. Each processing module is tailored to the unique computational requirements of each functional block. The result is a parallel processing system with a computational throughput that meets the most stringent KEW requirements.

2. Existing GN&C VLSI Chip Set

The existing GN&C VLSI chip set can be categorized into 4 subsets: executive processor, data processor, interconnection network, and signal processor. The VLSI design of all the chips have been completed. All of the chips have been fabricated or are being fabricated. The chips that have passed manufacturing testing were delivered to Georgia Tech. Most of the chips have also been laid out in the system board and run successfully as a system (refer to Testing and Evaluation report). All chip design databases and documents have been sent to Harris (see Table 1 and Table 2) to be converted to radiation hardened CMOS process (AHAT project).

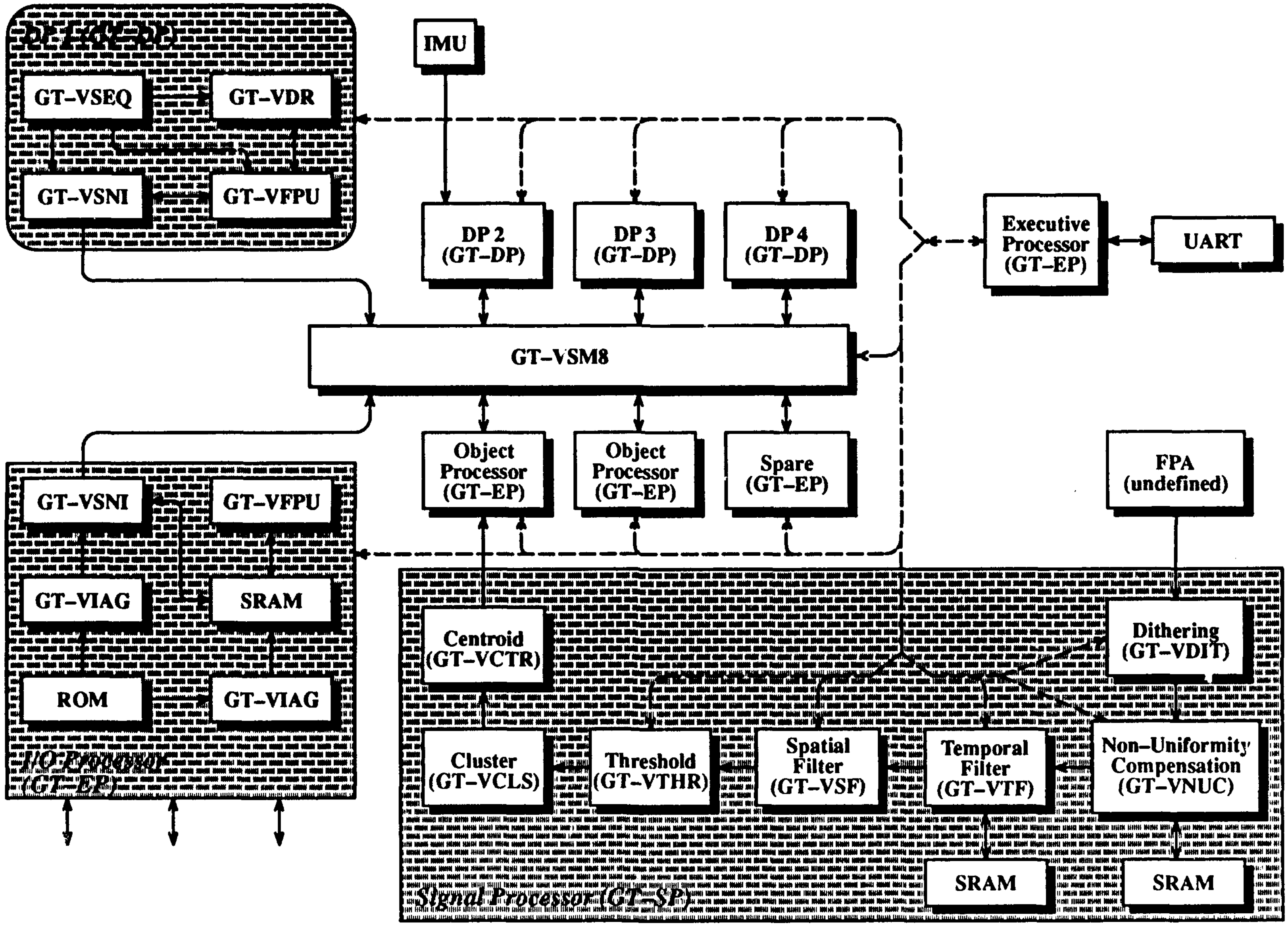


Figure 1: Architecture of the Georgia Tech GN&C Processor

Table 1. Georgia Tech Chip Set for AHAT

Design	DV Passed	Tape Delivered	Fabricated	Tested
GT–VFPU/1A	01/17/89	08/03/90	05/19/89	04/04/90
GT–VSNI	01/17/89	05/23/90	04/14/89	04/04/90
GT–VSM8	01/17/89	06/08/90	05/06/89	04/04/90
GT–VCTR	02/08/90	07/12/90	07/13/90	07/27/90
GT–VCLS	01/26/90	07/12/90	07/13/90	07/27/90
GT–VSF	09/12/89	07/19/90	07/13/90	07/27/90
GT–VTHR	12/11/90	02/15/91	03/01/91	03/08/91
GT–VDAG	02/22/91	02/25/91	05/01/91	
GT–VIAG	03/08/91	03/11/91	05/07/91	
GT–VNUC	07/23/91	07/05/91		
GT–VTF	07/24/91	08/01/91		
GT–VNUC (post DV)	07/23/91	08/16/91		
GT–VSF (v.2)	09/12/89	08/16/91	07/13/90	07/27/90

Table 2. Georgia Tech Documents Sent for AHAT

No.	Document / Software Item	Date Sent
1.	Georgia Tech GT–VFPU : VLSI Design Verification Document	05/15/90
2.	Georgia Tech GT–VSNI : VLSI Design Verification Document	05/23/90
3.	Georgia Tech GT–VSM8 : VLSI Design Verification Document	06/08/90
4.	Georgia Tech GT–VCTR : VLSI Design Verification Document	07/12/90
5.	Georgia Tech GT–VCLS : VLSI Design Verification Document	07/12/90
6.	Georgia Tech GT–VSF : VLSI Design Verification Document	07/19/90
7.	Instruction Address Generation GT–VIAG : Programming Model Document (v.1)	01/03/91
8.	GT–EP I/O Interface Specification : Note	01/17/91
9.	EP, SNI, SM8 Interconnection Specification : Note (v.1)	01/28/91
10.	Georgia Tech GT–VTHR : VLSI Design Verification Document	02/15/91
11.	Georgia Tech GT–VDAG : VLSI Design Verification Document	02/25/91
12.	Georgia Tech GT–VIAG : VLSI Design Verification Document	03/11/91
13.	GT–FPU Operating Speed Test Document	04/16/91
14.	Staggered Row Focal Plane Array Analysis Document	05/01/91
15.	GT–EP Pascal Compiler : Note (v.1), Source Code, and Program Examples	05/06/91
16.	Instruction Address Generation GT–VIAG : Programming Model Document (v.2)	06/07/91
17.	Georgia Tech GT–VNUC : VLSI Design Verification Document and User Guide	07/05/91

18.	EP, SNI, SM8 Interconnection Specification : Processor Design Document (revision to earlier Note sent on 01/28/91)	07/08/91
19.	GT–EP Pascal Compiler : Software User Document (revision to earlier Note sent on 05/06/91)	07/08/91
20.	GT–Seeker/Scene Emulator to GT–GN&C Processor Interface Document	07/19/91
21.	Georgia Tech GT–VTF : VLSI Design Verification Document and User Guide	08/01/91
22.	Data Address Generation GT–VDAG : Programming Model Document (v.2)	08/16/91

The following sections will describe the functionality of the three processor modules and the interconnection network. The characteristics of each fabricated chip are presented.

2.1. Executive Processor (GT–EP)

The executive processor provides overall executive control for the GN&C processor. Among the tasks to be executed by the executive processor are initialization of the GT–DP and GT–SP processors, overall system consistency checks, flight phase/mode control, target tracking functions, and computational support for other devices such as the IMU and control valves. To perform these executive functions, the GT–EP processor needs to have access to considerably larger amounts of instruction and data memory than the GT–DP processor. In addition, the GT–EP processor must handle real–time tasks and event scheduling in which fast interrupt response capability is critical. Furthermore, the GT–EP must be able to support the object processing requirements. All of this functionality has been incorporated in the GT–EP processor. A total of 5 GT–EP processors are used on the Georgia Tech GN&C processor shown in Figure 1: one as the executive processor, one as an I/O processor, and three as object processors. These numbers can be varied to meet specific requirements.

As shown in Figure 2, the GT–EP processor consists of six functional units: Instruction Memory, Data Memory, Instruction Address Generation, Data Address Generation, Arithmetic Logic Unit, and Network Interface. The arithmetic logic unit uses the GT–VFPU chip developed for the GT–DP processor (see section 2.2). The network interface uses the GT–VSNI which is connected to an 8–point fully connected crossbar switch (see section 2.3).

Instruction execution for the GT–EP processor is classified as *user* or *kernel*. In user mode, the instruction address and data address are checked against a pre–specified range. An address out of range violation will cause an interrupt to an exception handling routine. This feature provides extra protection for the GT–EP processor to service real–time devices in a real–time environment. Furthermore, instruction execution for the GT–EP processor is very deterministic, permitting the GT–EP processor to work under stringent timing constraints.

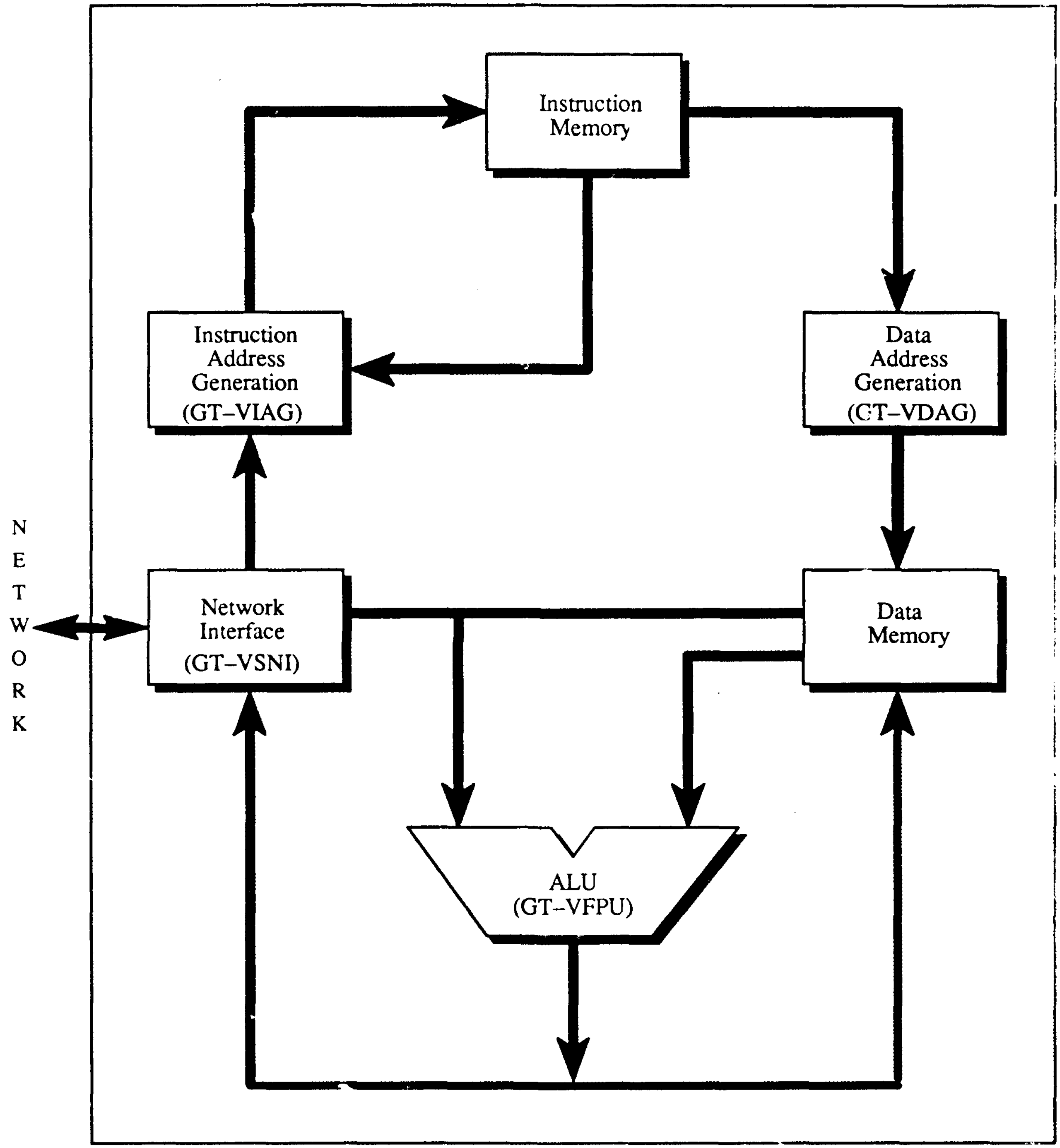


Figure 2. GT–EP Functional Modules

Two custom VLSI chips are required to implement the instruction address generation unit and the data address generation unit. These two VLSI chips are designated GT–VIAG and GT–VDAG. Commercially available EPROM and RAM chips are used for instruction and data memory in the GT–EP processor. Using external memory for instruction and data memory, instead of designing it into the VLSI chips, allows flexible memory configurations based on the final system requirements. A standard memory interface is incorporated into the GT–VDAG and GT–VIAG chips to allow a direct interface with commercially available EPROMs and RAMs.

A more detail description of the GT–EP architecture is documented in paper [33]. The paper was published and presented in the International Symposium on Computer Architecture, May 27–30, Ontario, Canada. The response was very positive and encouraging. The following subsections describe briefly about the primary functions of each chip (GT–VIAG, GT–VDAG, and GT–VFPU). The characteristics of the three chips are given in Table 3. An overview of the architecture, algorithm and features of each chip can be found in last year annual report [34].

Table 3. Key Parameters of GT–VIAG, GT–VDAG, and GT–VFPU VLSI Chips

Chip Name	Die Size (mil x mil)	Power (W)	Number of Transistors	Package	Technology
GT–VIAG	417 x 425	1.18	65,190	224 CPGA	HP CMOS 1.0 u
GT–VDAG	415 x 410	0.95	54,569	224 CPGA	HP CMOS 1.0 u
GT–VFPU	379x363	5.10	53,000	144 CPGA	NSC CMOS 1.25 u

2.1.1. GT–VIAG (Instruction Address Generation)

The primary function of this chip is to generate instruction memory address of the next instruction. It also generates and controls the instruction fields for the ALU and I/O operations. In addition GT–VIAG chip supports prioritized interrupts and multitasking. A more detailed description of this chip can be found in [17] – [20].

The chip has been fabricated and currently being tested at the IC level (manufacturing test). Two engineering samples are at Georgia Tech to be used for system level testing.

2.1.2. GT–VDAG (Data Address Generation)

The GT–VDAG chip is used to generate two address fields for operand fetches and one address field for result store. Besides direct addressing mode, the chip supports post–index addressing, for accessing arrays with constant strides, at a rate of one cycle per array element. Relative addressing is also supported to ease local variable accesses and parameter accesses for recursive procedures. Built–in automatic operand–dependency check circuitry alleviates the need to insert NOPS at the end of every basic program block. A more detailed description of this chip can be found in [13] – [16].

The chip has been fabricated and currently being tested at the IC level (manufacturing test). Two engineering samples are at Georgia Tech to be used for system level testing.

2.1.3. GT–VFPU (Fixed/Floating Point Unit)

The actual data computation is performed in this chip. Three data types are supported: floating–point, fixed–point, and bit–field. The floating–point data type is a single precision, 32–bit number, in IEEE floating–point format. The fixed–point data type is a 23–bit, signed–magnitude number. The bit–field data type is an unformatted 32–bit number. The GT–VFPU chip operates in three pipeline stages: 1 stage for operand fetches, 1 stage for data computation, and 1 stage for a result store. An automatic operand–dependency scheme is used to control the internal feedback paths. This feature enables the GT–EP processor to execute scalar computations efficiently. A more detailed description of this chip can be found in [5] – [8]. The GT–VFPU chip is used by the GT–DP processor as well.

The chip has been fabricated and successfully tested at the IC level (manufacturing test) and system level. Eight (8) working parts are at Georgia Tech.

2.2. Data Processor (GT–DP)

The data processor is used to perform numerically intensive tasks for guidance, navigation, and control of the KEW interceptor. This type of computation is floating–point intensive and requires very high scalar throughput. These computational tasks do not require large amounts of instruction and data memory (less than 1 Kbytes). The Georgia Tech Data Processor was designed to meet these requirements. Four GT–DP processors are shown in . The number can be changed up or down to meet specific KEW requirements.

As shown in Figure 3 the GT–DP processor consists of four functional blocks: Instruction Control (GT–VSEQ), Data Control (GT–VDR), Arithmetic Control (GT–VFPU), and Communication Control (GT–VSNI). Table 4 shows the characteristics of GT–VSEQ and GT–VDR chips. The following subsections describe briefly the primary functions of GT–VSEQ and GT–VDR. GT–VFPU has been described in the Executive Processor section (section 2.1). GT–VSNI will be described in the Interconnection Network section (section 2.3).

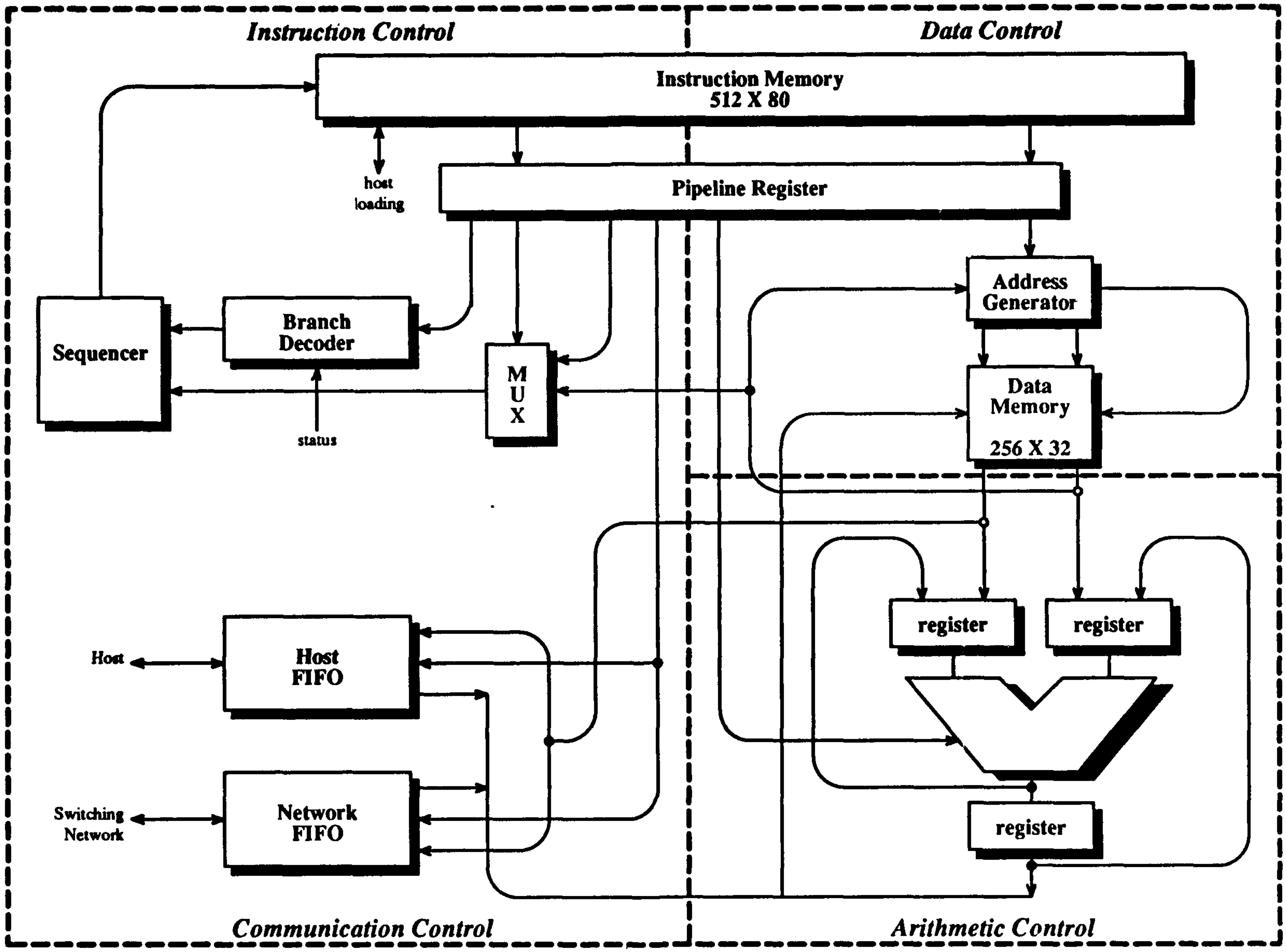


Figure 3: GT–DP Processor Architecture

Table 4. Key Parameters of GT–VSEQ and GT–VDR VLSI Chips

Chip Name	Die Size (mil x mil)	Power (W)	Number of Transistors	Package	Technology
GT–VSEQ	371 x 410	0.70	131,000	100 CPGA	NSC CMOS 1.5 u
GT–VDR	510 x 450	2.10	242,000	180 CPGA	US2 CMOS 1.5 u

2.2.1. GT–VSEQ (Sequencer)

The Instruction Control Unit (GT–VSEQ) is mainly responsible for the generation of instruction addresses. It receives status flags from the Arithmetic Control Unit (GT–VFPU) and appropriately determines the next instruction address. It facilitates branch–lookahead for efficient pipelined arithmetic instruction execution. A more detailed description of this chip can be found in [9] and [10].

The chip has been fabricated and successfully tested at the IC level (manufacturing test) and system level. Five (5) working parts are at Georgia Tech.

2.2.2. GT–VDR (Dataram)

In each computing cycle, the Data Control Unit (GT–VDR) supplies two operands to the Arithmetic Control Unit (GT–VFPU). In addition, it receives a result from the Arithmetic Control Unit for storage. Three addressing modes are supported: direct, indexing, and post–indexing. The direct addressing mode directly specifies data addresses for two operands in the data memory. The indexing mode specifies 1 of 16 index registers to add to the data address value from the instruction memory. The post–indexing mode increments the value of the index at the end of the computing cycle. A more detailed description of this chip can be found in [11] and [12].

The chip has been fabricated and successfully tested at the IC level (manufacturing test) and system level. Five (5) working parts are at Georgia Tech.

2.2.3. GT–VFPU (Fixed/Floating Point Unit)

The same GT–VFPU chip used in GT–EP is being used in GT–DP. Please refer to section 2.1.3.

2.3. Interconnection Network

Two VLSI chips are designed to provide the fully–connected communication among various processor modules (GT–VSM8) and proper network interfacing at each processor module (GT–VSNI). Table 5 shows the characteristics of GT–VSNI and GT–VSM8 VLSI Chips.

Table 5. Key Parameters of GT–VSNI and GT–VSM8 VLSI Chips

Chip Name	Die Size (mil x mil)	Power (W)	Number of Transistors	Package	Technology
GT–VSNI	301x272	0.60	54,000	120 CPGA	NSC CMOS 1.25 u
GT–VSM8	338 x 326	0.81	49,967	100 CPGA	NSC CMOS 1.25 u

2.3.1. GT–VSNI (Serial Network Interface)

The GT–VSNI chip is used both in the executive processor (GT–EP) and data processor (GT–DP). The GT–VSNI chip is the Communication Control Unit, which is used to control the communication between the GT–DP (or GT–EP) processor and other processor modules connected to an 8–point fully–connected network. The GT–VSNI chip consists of two pairs of 32–word FIFOs. One pair of FIFOs is used to communicate with other processors through the crossbar network. Another pair is used to communicate with the executive processor. Data to the crossbar network is transmitted serially in 32–bit data and 7–bit parity packets. The 7–bit parity performs a single bit error correction and double bit error detection on packets transmitted across the network. A more detailed description of this chip can be found in [1] and [2].

The chip has been fabricated and successfully tested at the IC level (manufacturing test) and system level. Six (6) working parts are at Georgia Tech.

2.3.2. GT–VSM8 (Switch Matrix 8x8)

This chip allows a fully–connected communication among various processor modules in a closely coupled interconnection network. It is designed as an 8–point crossbar/matrix switch. A more detailed description of this chip can be found in [3] and [4].

The chip has been fabricated and successfully tested at the IC level (manufacturing test) and system level. Eight (8) working parts are at Georgia Tech.

2.4. Signal Processor (GT–SP)

The signal processor was designed to process infrared images from a focal plane array (FPA) with 128x128 pixel resolution at a rate of 100 frames per second. Each pixel is assumed to have a 12–bit resolution with a dynamic range of 16 bits. The signal processor performs various filtering operations on the pixel data before clustering them into objects for target tracking and discrimination. The signal processor is decomposed into 8 functional blocks for VLSI implementation (see Figure 1). The first functional block is the FPA interface (not fully defined yet) which is used to link the signal processor and the focal plane array. The next block is dithering (GT–VDIT) for clutter rejection. This operation and its design development are discussed in the next generation signal processor (GT–SP/2) section.

In general, the existing signal processor processes pixel information and outputs clusters of targets for tracking and discrimination. At the pixel level, non–uniformity compensation (GT–VNUC) is used to compensate the non–linearity characteristics of the infrared detector response. Temporal filtering (GT–VTF) performs time averaging of pixel values across frames to reduce random noise and smearing of the images due to jittering of the FPA. Spatial filtering (GT–VSF) reduces the effect of spatial noise across pixels and enhances the image contrast. Thresholding (GT–VTHR) suppresses noise and increases target discrimination by cutting out pixels that exceed a constant or calculated threshold. Clustering (GT–VCLS) allocates adjacent pixels and clusters them into targets. Finally, centroiding (GT–VCTR) calculates the intensity and area centroid of each target in the field of view. These functions combined require a computational throughput in excess of 600 million operations per second (MOPS).

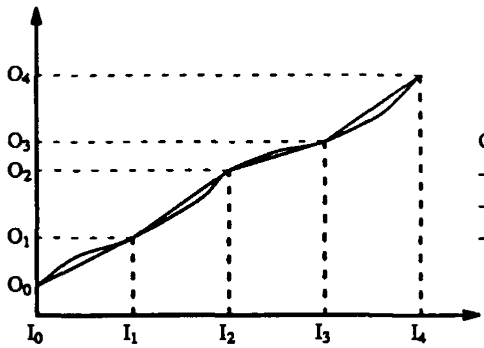
All of the chips have been fabricated and tested, except GT–VNUC and GT–VTF which are currently being fabricated at Hewlett Packard. Table 6 shows some characteristics of the GT–SP VLSI chips. The following subsections describe briefly about the primary functions and algorithm of each chip.

Table 6. Key Parameters of GT–SP VLSI Chips

Chip Name	Die Size (mil x mil)	Power (W)	Number of Transistors	Package	Technology
GT–VNUC	399 x 403	1.07	60,000	180 CPGA	HP CMOS 1.0 u
GT–VTF	418 x 421	0.93	81,253	180 CPGA	HP CMOS 1.0 u
GT–VSF	335 x 311	0.80	40,000	100 CPGA	HP CMOS 1.0 u
GT–VTHR	405 x 400	0.85	123,807	100 CPGA	HP CMOS 1.0 u
GT–VCLS	334 x 390	1.90	67,000	84 CPGA	HP CMOS 1.0 u
GT–VCTR	395 x 396	1.08	117,000	120 CPGA	HP CMOS 1.0 u

2.4.1. GT–VNUC (Non–Uniformity Compensation)

The GT–VNUC is used to compensate nonlinear detector characteristics in the FPA. The response of each detector is compensated with 4 piecewise linear segments. During calibration, the FPA is irradiated with five known sources. Based on the FPA response, 4 linear segments are constructed for each pixel (see Figure 4). If the output response of a pixel is not monotonically increasing with monotonically increasing stimuli, the pixel is marked as a bad pixel. During normal operation, each pixel value is mapped from one of the four linear segments to a common desired response. The output reponse of a bad pixel is taken from the response of the previous pixel. The chip is currently fabricated at Hewlett Packard.



Compensation:
- Sample output response: O_r
- Find n such that $O_n < O_r < O_{n+1}$
- Calculate $I_c = Pw(O_r,n)$

Calibration:
- Apply 5 known sources: I_n, n=[0,4]
- Sample output response: O_n, n=[0,4]
- Obtain 4 piecewise linear functions:

$$Pw(O_r,n) = \frac{(I_{n+1} - I_n)}{(O_{n+1} - O_n)} (O_r - O_n) + I_n \quad for\ n = [0,3]$$

Storage Requirement:
5*128*128*16)= 1.3 M bits

Figure 4. Non–Uniformity Compensation

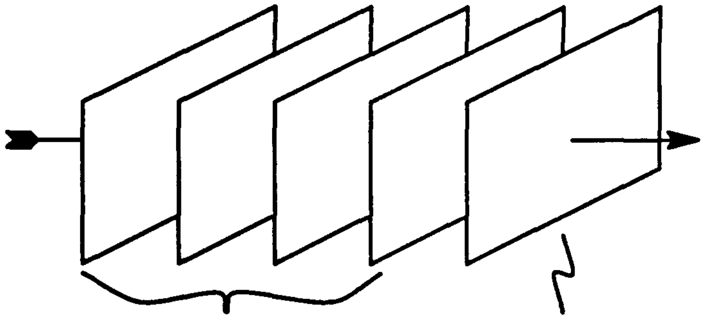
2.4.2. GT–VTF (Temporal Filtering)

The GT–VTF performs time averaging of pixel values across frames. It is used to reduce random noise across frames as well as smearing of images due to a jittering motion on the FPA. The GT–VTF is a fourth order temporal filter which computes the output pixel response based on the current input pixel, the four previous input pixels, and the four previous output pixels. Nine programmable coefficients are used to program the GT–VTF to conform to desired response characteristics. Based on the values of the coefficients, the GT–VTF functions as a FIR or IIR filter. The functionality of the GT–VTF is illustrated in Figure 5. The chip is currently fabricated at Hewlett Packard.

2.4.3. GT–VSF (Spatial Filtering)

The GT–VSF performs filter operations based on the current pixel value and the immediate eight surrounding pixels. The GT–VSF is used to reduce the effects of spatial noise and to enhance/ reduce the contrast of the FPA image. Four filter coefficients are used to configure the GT–VSF response characteristics. The four coefficients represent the weighting factors for the current pixel, the four diagonal pixels, the two horizontal pixels, and the two vertical pixels. The functionality of the GT–VSF is illustrated in Figure 6.

The chip has been fabricated and successfully tested at the IC level (manufacturing test at 330–ns pixel clock) and system level. Fifty five (55) working parts are at Georgia Tech.

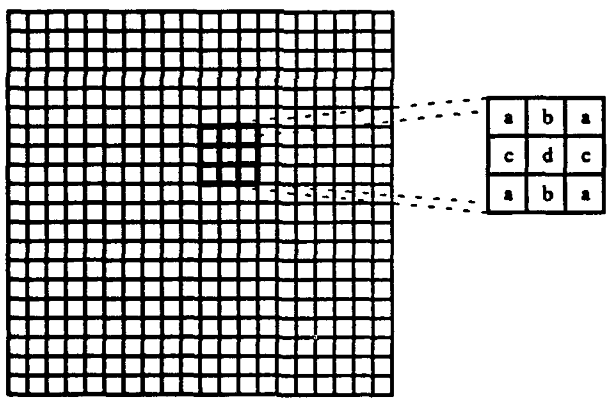


$$PixelP_{ij}(t_n) = \frac{\sum_{k=0}^{4} b_k Z^{n-k} P_{ij}}{1 + \sum_{k=1}^{4} a_k Z^{n-k} P_{ij}}$$

Storage Requirements: 4*128*128*16 = 1 M bits

Figure 5. Temporal Filtering

9 points, 2D, FIR Filter

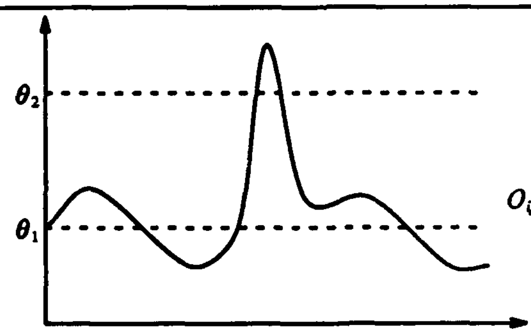


$$\begin{aligned} O_{i,j} = {} & a(P_{i-1,j-1} + P_{i-1,j+1} + P_{i+1,j-1} + P_{i+1,j+1}) + \\ & b(P_{i-1,j} + P_{i+1,j}) + \\ & c(P_{i,j-1} + P_{i,j+1}) + \\ & d(P_{i,j}) \end{aligned}$$

Figure 6. Spatial Filtering

2.4.4. GT–VTHR (Thresholding)

The GT–VTHR is used to suppress noise by cutting out pixels that fall outside the range between a lower and upper threshold. The upper threshold is a constant. The lower threshold is either a constant or a computed value. The computed lower threshold can either be computed as an adjusted threshold or adaptive threshold. Adjusted thresholding allows the lower threshold value to be dynamically adjusted according to the number of pixels passed by the GT–VTHR on the previous frame. Adaptive thresholding computes the lower threshold based on a statistical average of the 8 pixels which surround the pixel under evaluation. These thresholding schemes are illustrated in Figure 7.



$$O_{ij} = \begin{cases} I_{ij}, \ \forall \theta_1 \leq I_{ij} \leq \theta_2 \\ \theta, \ otherwise \end{cases}$$

1. Simple thresholding: θ_1 *&* θ_2 *constants*

2. Adjusted thresholding: $\theta_2 = constant, \theta 1 = Z^{-1}\,\hat{o}\,\delta$

$$\hat{o} = \begin{cases} +, \ \#P > N_1 \\ -, \ \#P > N_2 \\ nop, \ N_1 \leq \ \#P \leq N_2 \end{cases} \qquad \begin{array}{l} Z^{-1} : frame\ delay \\ \#P : number\ of\ pixels\ passed \\ N_1, N_2 : programmableconstants \end{array}$$

3. Adjusted thresholding: $\theta_2 = constant, \theta 1 = k_1 E + k_2 L_1 + k_3; k_1 \geq 0$

E_1 *= average value over neighboring 8 pixels*

L_1 *= absolute sum of the difference between* E_1 *& neighboring 8 pixels*

Figure 7. Thresholding

The chip has been fabricated and successfully tested at the IC level (manufacturing test at 350–ns pixel clock) and system level. Fifty one (51) working parts are at Georgia Tech.

2.4.5. GT–VCLS (Clustering)

The GT–VCLS groups adjacent pixels with non–zero intensity into clusters. Two non–zero pixels are assigned to a cluster if the distance between them is no more then 1, e.g. pixel p(i,j) and p(k,l) are adjacent if (|i–k|<2) and (|j–l|<2). Scanning the pixels from left to right and top to bottom, a cluster is complete if no pixel in a row is adjacent to any pixel in the cluster. A parallel scheme using a 128 entry associative search algorithm is used to identify and group the clusters in the field of view of the FPA. The GT–VCLS can handle up to a maximum of 4096 objects.

The chip has been fabricated and successfully tested at the IC level (manufacturing test at 300–ns pixel clock) and system level. Fifty four (54) working parts are at Georgia Tech.

2.4.6. GT–VCTR (Centroiding)

For each cluster in the field of view identified by the GT–VCLS, the GT–VCTR calculates the total area (#A), the total intensity (#I), the area centroid(A_x,A_y), and the intensisty centroid

(I_x, I_y). #A is calculated by counting the number of pixels in cluster P_k,

$$\#A = \sum_{i,j \in P_k} 1$$

#I is calculated from the summation of the intensity of the pixels in the cluster P_k,

$$\#I = \sum_{i,j \in P_k} I(i,j)$$

The area centroid is calculated from

$$A_x = \frac{\sum_{i,j \in P_k} x(i,j)}{\#A}, \text{ and}$$

$$A_y = \frac{\sum_{i,j \in P_k} y(i,j)}{\#A},$$

where x(i,j) and y(i,j) are the x and y coordinates of the pixels in cluster P_k. The intensity centroid is computed from

$$I_x = \frac{\sum_{i,j \in P_k} I(i,j)\, x(i,j)}{\#I}, \text{ and}$$

$$I_y = \frac{\sum_{i,j \in P_k} I(i,j)\, y(i,j)}{\#I}.$$

The chip has been fabricated and successfully tested at the IC level (manufacturing test at 330–ns pixel clock) and system level. Fifty five (55) working parts are at Georgia Tech.

3. Next Generation of Integrated Circuits

3.1. Introduction

3.1.1. History

The VLSI development effort at DETL has been underway for about six years. During that time, two complete processors have been developed and tested (the data processor and executive processor), and portions of a third processor are under test (the signal processor). The basic philosophy has been to implement in silicon certain functions known to be useful, either because of previous designs or because of a problem specification. The result has been a highly successful series of chips and chipsets which have to date satisfied all requirements for function and performance.

An inevitable consequence of this philosophy is the desire to improve on working designs once areas of improvement become apparent. It is important to understand that "New and Improved" in the context of VLSI design does not mean that older designs are not useful or successful. Rather, it means that access to smaller and faster fabrication technologies and more sophisticated software tools enables a designer to put in features that previously would not have fit onto one chip.

3.1.2. Objectives

The objectives of the ongoing design effort is simple. How can we make what we have designed better? Are there any additional requirements that must be met or problems that must be solved? The answers to these questions come by examining our requirements—both the original set and the set that has emerged in the past six years.

3.1.3. Requirements

The original requirements for data processing have remained largely unchanged. Two 10 MHz processor modules have been designed and tested (or are in testing), one with a small chip count and the other with larger memory capacities. The emerging requirement is for double–precision floating point capability. This could be supported very crudely by the existing Executive Processor, but with an enormous speed penalty. This has led to the development of a double precision FPU (GT–VFPU/3) which is described in Section 3.2.1. Another incremental improvement would be to integrate the three chips that comprise the executive processor into one chip. This would reduce chip count and power consumption and possibly increase system speed. This is discussed in Section 3.2.2.

The interconnection among processors is the key to efficient parallel programming. Two chips currently support interconnection. They are the crossbar itself (GT–VSM8) and an interface with the crossbar (GT–VSNI). The crossbar supports full interconnection between processors so that any processor can be connected in a non–blocking manner to any other processor. These chips have been fabricated and tested, and a parallel version is under development. This is discussed in Section 3.3.

Many new requirements have emerged since design work began on the signal processing chipset. All of the chips are either out for fabrication or back in and installed on functioning boards, and so no modification is possible for these existing chips. Some features may be installed on future chips, and it is these features that are of interest here.

One feature that has a sizable impact on hardware is the use of staggered pixel rows. This design has been proposed in some focal plane arrays. Pixel geometry is implicitly assumed by the hardware, and so some of the chips (Spatial Filtering, Thresholding, and Clustering) must be modified if they are to handle staggered pixels. The geometry of staggered pixels is illustrated in Figure 8.

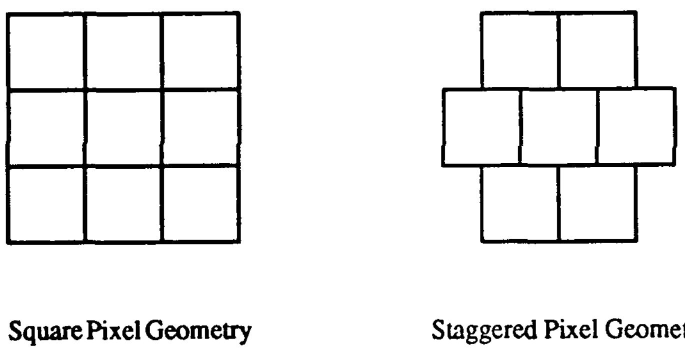


Figure 8: Square versus Staggered Pixel Geometry

One technique that may be used to speed frame processing is the use of windowing. The assumption is that if the processor only has to manage a small subset of each frame, entire frames can be processed much faster. For example, the current processor can process a 128 X 128 frame at 100 frames per second. If the window size were 64 X 64 pixels, the frame rate could potentially be increased by a factor of four, since only one–fourth the pixel are being processed. This windowing capability is not built into current designs.

Finally, some additional signal processing algorithms have been identified as candidates for direct silicon implementation. These include dithering (which compensates for a scanning motion by the seeker), object feature extraction, and sensor fusion. The latter would combine information from a variety of sensors (e.g. visible, near and far infrared and laser radar).

These and various other issues are discussed in the section concerning ongoing signal processor design efforts, Section 3.4.

3.2. Executive Processor

3.2.1. GT–VFPU/3

There are three versions of this chip currently. The first, GT_VFPU/1, has been fabricated by National Semiconductor and has been successfully installed in the Data Processor and Executive Processor. The second version ("/2") was part of an unsuccessful attempt by Harris Semiconductor to fab in the now–defunct Gamma 3 fabline. The third version (GT–VFPU/1a) is the version in use by the AHAT program. It contains very minor modifications for use in the Executive Processor.

Two shortcomings have been identified with the current FPU (GT–VFPU/1). First, it does not directly support double precision floating point instructions. Second, its interface with the GT–EP could be made more efficient. For these reasons, work has begun on a "next generation" floating point unit, GT–VFPU/3.

The existing FPU design has been extensively modified and will incorporate the following features. First, all integer operations will be performed in two's complement. (The previous chip used sign–magnitude.) Second, it will support 8, 16, and 32–bit integers (signed and unsigned) and 32 and 64–bit floating point numbers. The floating point format will be the IEEE standard. This range of supported types will simplify future compiler development and insure conformity to existing standards. A list of opcodes is shown in Table 7.

Table 7: Complete Opcode List for FPU/3

	8–bit Integer		16–bit Integer		32–bit Integer		Float	
	Signed	Un-signed	Signed	Un-signed	Signed	Un-signed	Single (32)	Double (64)
op[6:4] op[3:0]	000	001	010	011	100	101	110	111
0000	ADD	ADD	ADD	ADD	ADD	ADD	ADD	ADD
0001	SUB	SUB	SUB	SUB	SUB	SUB	SUB	SUB
0010	MULT	MULT	MULT	MULT	MULT	MULT	MULT	MULT
0011	RSUB	RSUB	RSUB	RSUB	RSUB	RSUB	RSUB	RSUB
0100		AND		AND	PACK EXP	AND	INV SEED	AND
0101		OR		OR		OR	ROUND	OR
0110		XOR		XOR		XOR	UPACK EXP	XOR
0111		NOT		NOT		NOT	UPACK MANT	NOT

1000		SHL		SHL		SHL	ROOT EXP	SHL
1001		SHR		SHR		SHR	ROOT MANT	SHR
1010	ASR	PASS	ASR	PASS	ASR	PASS	ROOT SEED	PASS
1011		ROL		ROL		ROL		ROL
1100		ROR		ROR		ROR		ROR
1101			TO8S	TO8U	TO8S	TO8U	TO32S	TO32S
1110	TO16S	TO16U			TO16S	TO16U	TO32U	TO32U
1111	TO32S	TO32U	TO32U	TO32S	TODBL	TODBL	TODBL	TOSGL

The opcodes at the bottom of the table are conversions between formats. There may be some additional opcodes implemented eventually. A simplified block diagram of the chip is shown in Figure 9.

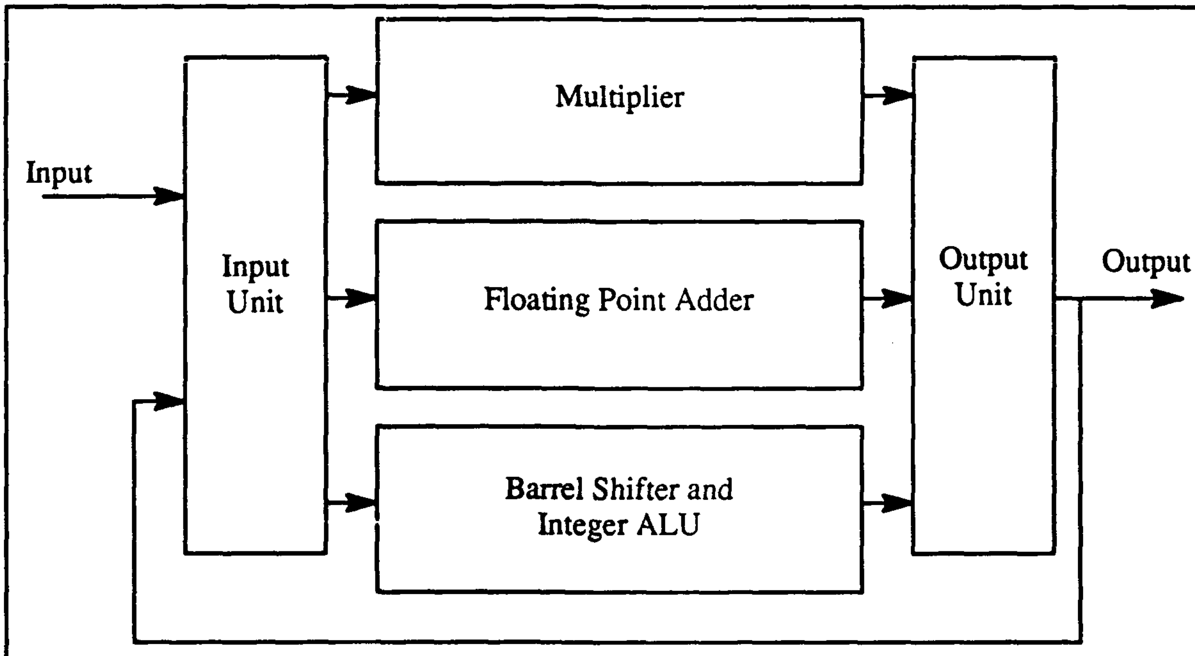


Figure 9: Block Diagram of GT–VFPU/3

It is currently believed that the chip will be small enough to be manufacturable and fast enough to run at 10 MHz. Its 10 MHz double–precision capability will make it a strong candidate to replace both the GT–FPP and GT–FPX boards in the PFP. Using Matra's 1.0 micron fabline, the current chip size is 440 mils by 410 mils without the pads (which will add at least 50 mils to each side) and the current cycle time is 116 ns, which corresponds to a clock speed of 8.6 MHz. Improvements in clock speed are expected to drive this figure up to 10 MHz.

The chip should be ready for Design Verification by the end of calendar year 1991.

3.2.2. GT–VEP (IAG / DAG / FPU Integration)

It has been known for some time that the transistor count of the three components of the EP are so small that it is conceivable all three could fit on one chip. The largest technical hurdles are the pin count (which would be extremely high) and a sufficiently small and efficient fabline that can be used by the silicon compiler.

The pin count would be in the vicinity of 300 I/O's, plus power and ground, as shown in Table 8. The "Misc" entry includes I/O handshaking, external interrupts and other such signals. The 64–bit RF bus assumes that a double–precision ALU is available. Further work is needed to reduce this figure.

Bus	Pins Needed
Instruction	136
Pc	26
RF_adr	26
SF_adr	26
RF	64
Misc	25
Power/Gnd	70
Total	373

Table 8: I/O Count for Combined EP Chipset

A 0.8 micron fabline may be available soon on the compiler, and this will be pursued as soon as it is available. It is not clear that 0.8 microns will be small enough to enable a design to be manufacturable. A block diagram of such a system is shown in Figure 10.

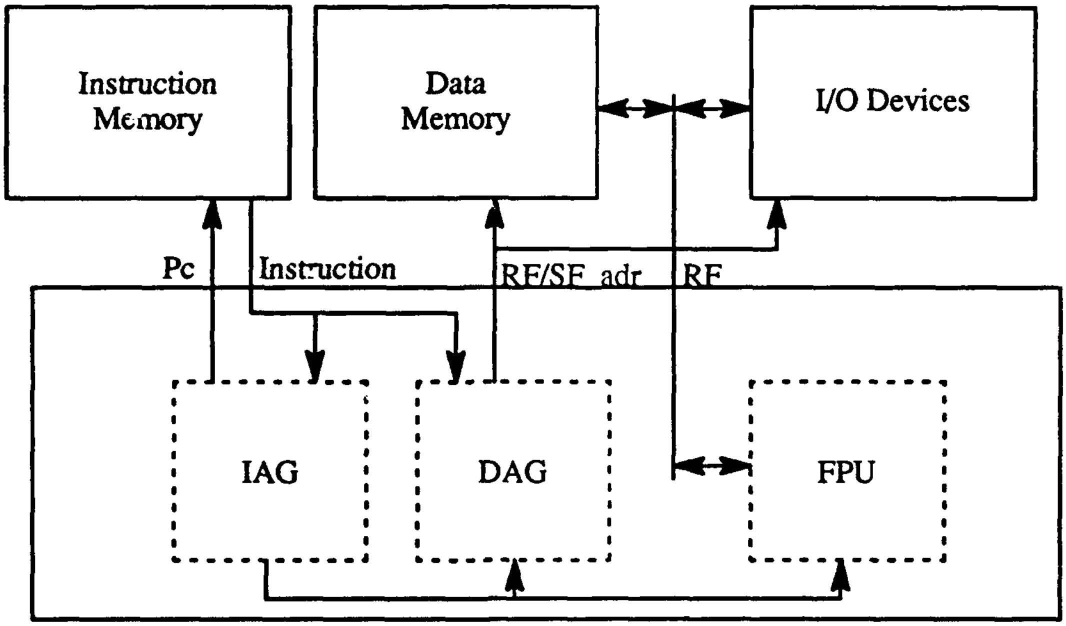


Figure 10: Single–Chip Executive Processor

There are two potential payoffs. First, system design is made much simpler because one chip replaces three. Second, a faster clock speed may be possible because more functions are contained on chip. (Going from one chip to another is extremely costly in terms of speed.)

Another, less ambitious option is to modify the existing IAG and DAG chips and retain the three–chip configuration of the EP. The bus widths would be expanded from 26 bits to 32 bits and ideally the speed would improve. This would then entail the development of two new chips (besides the FPU/3), which would be designated GT–VIAG/2 and GT–VDAG/2.

3.3. Interconnection Network

3.3.1. GT–VPNI

The functional block diagram of the GT–VPNI chip is shown in Figure 11. The Network Controller arbitrates the data communication between the EP–Bus and the Interconnection Network. Data from the GT–EP processor can be sent to any other processors connected through the Interconnection Network. Three types of message passing protocols will be supported: point–to–point, muticast, and interrupt based. A point–to–point channel is a data channel between two processors, one as a sender the another as a receiver. To establish a point–to–point channel, a processor sends an interrupt message to another processor signaling the desire to receive or send. A multicast channel involves more than two processors. There may be multiple senders and multiple receivers involved in a multicast transfer cycles. Synchronization is achieved when all the partipaticipating processors indicate that they are ready to send and receive.

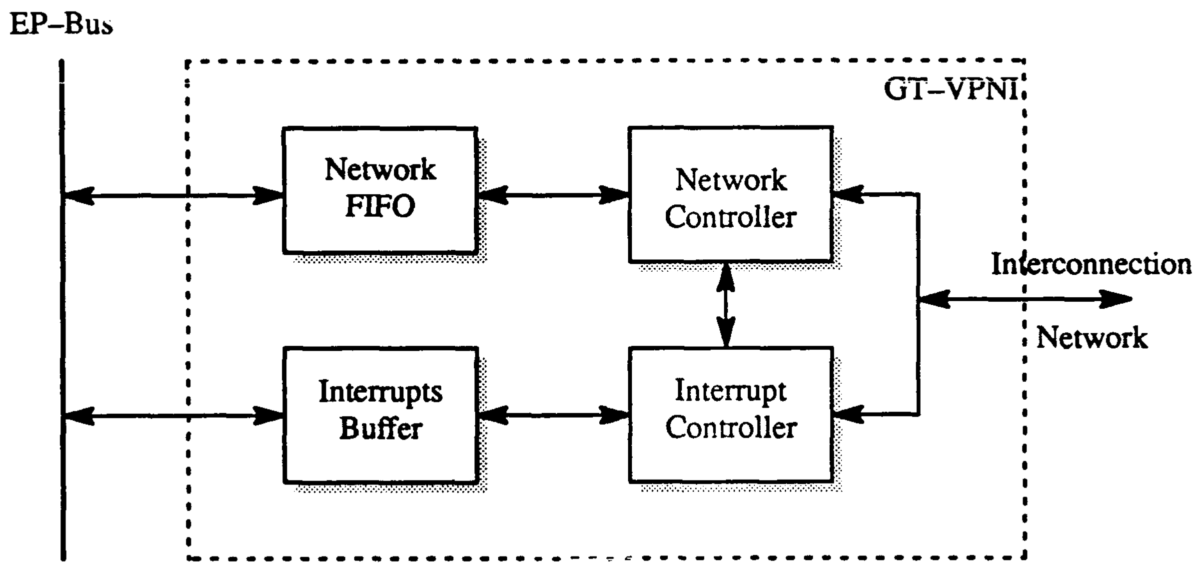


Figure 11. GT–VPNI Functional Block Diagram

A comparison of the basic features of the GT–VPNI and the GT–VSNI is shown in Table 9. The basic structure of the GT–VPNI chip is similar to the GT–VSNI. The new GT–VPNI replaces the host communication port with an interrupt controller. With this arrangement, any processor on the system can serve as a host. The interrupt mechanism allows the processors to work in a closely coupled manner to form an effective distributive parallel processing system.

Table 9. Comparision of GT–VPNI and GT–VSNI

Description	GT–VPNI	GT–VSNI
EP–Bus Data	64–bit	32–bit
Network Data	10–bit (2–bit ECC)	1–bit
Interprocessor Interrupt	Supported	Not supported
Transfer Rate	40 MB/s	2 MB/s
Communication Protocol	Point–to–point/ Multicast	Fixed Sequence

3.3.2. GT–VSM8/2

The existing GT–VSM8 chip supports bit serial transfer operations between eight processor elements. The transfer rate between any processor pair is 20 Mbits/s. Excluding the error correction code, the effective transfer rate is actually 16 Mbits/s. The effective throughput is much higher if one sender broadcasts data to multiple receivers or multiple senders are active in a single transfer cycle.

The GT–VSM8/2 will improve the existing GT–VSM8 chip in two major areas. The first area is in transfer speed and the second area in improved capabilities. The speed improvement will come from the speeding up of the clock speed and transfering data in parallel. A factor of 2 improvement can be expected in the increase of clock speed. Another factor of 10 improvement is achieved by going to byte wide transfers (with two additional parity bits). This would yield a total expected performance improvement of a factor of 20.

In addition, the GT–VSM8/2 chip will be equipped to handle the three communication protocols mentioned in the previous section. The new implementation of the GT–VSM8/2 chip essentially distributes the communication sequencing control to each individual GT–VPNI chips. This results in a more flexible network that can betterly deal with non–determistic types of communication transactions between the processor elements.

The block diagram of the GT–VSM8/2 chip is shown in Figure 12. The pincount of the chip might become an issue. Using 10–bit data paths for the GT–VPNI chips would result in 160 pins required for the crossbar switch data lines. An additional 12 pins would be required to setup the crossbar switch control lines. The Muticast and Interrupt modules may require an additional 120 pins. The pin count could be reduced by combining the pins required for the three functional modules. This issue will be resolved as the design of the chip progresses.

In addition to the three modules, the GT–VSM8 might require a host port to set up the operating modes of the modules. An alternative approach would be to program the modules through

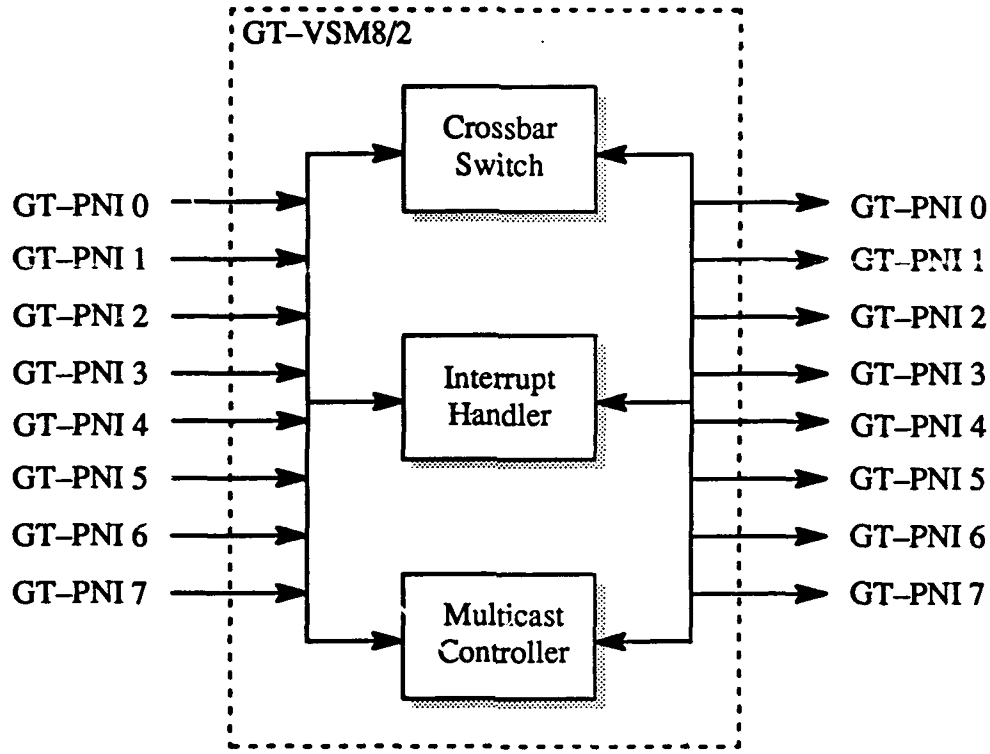


Figure 12. GT–VSM8/2 Functional Block Diagram

the GT–PNTC. This will save the pin count and increase the flexibility, because the GT–VSM8 can then be programmed from any processor in the system, at the expense of a slight increase in the chip complexity.

The design of the GT–VSM8/2 chip will strive to maximize the performance and the capbilities of the interconnection network in the following areas:

Data Transfer Speed : the aggregate data bandwidth between a sender and a receiver,

Network Latency : the time required for data to travel from a sender to a receiver,

Channel Setup Time: the time required to establish a connection,

Flexibility: the ability to support flexible communication protocols between processors,

Connectivity: the ability to connect to all processors without going through an intermediate processor(s), Multicast: the ability for a sender to transmit data to multiple processors, including the optiopn to broadcast to all processors including to itself, in a single data transfer cycle.

3.3.3. Development Status

The development schedule for the GT–VPNI and the GT–VSM8/2 chips is given in the later section along with the development schedule for the rest of the next generation chip set.

The development effort for the two chips had not been started. The testing and evaluation of the existing GT–VSNI and GT–VSM8 chips, which is still in progress, will provide valuable insights into the new design. Ultimately, the most important issue is whether the existing chipset is equipped to handle the next generation high performance GN&C processors and in what area can the next generation chipset improve to make a better and more effective GN&C processor.

3.4. Signal Processor

3.4.1. GT–VSF/2, GT–VTHR/2, GT–VTF/2, GT–VNUC/2

All four of these existing SP chips will need to be modified to handle staggered pixel rows, windowing, and a completely variable frame size. (Many support the variable frame size already.) All of these functions may be carried out by a Programmable Signal Processor (GT–VPSP), which is discussed in Section 3.4.7.

3.4.2. GT–VCLS/2 (Clustering)

As above, the Clustering chip will be modified to include handling of staggered rows, windowing, and variable frame size. The clustering function itself is so unique that its implementation requires special–purpose hardware, and so the GT–VPSP cannot be used.

The use of staggered pixels would dramatically change the connectivity between pixels. It is a relatively simple feature to add. Variable frame size (and windowing) is more difficult to support.

3.4.3. GT–VCTR/2 (Centroiding)

The next generation centroiding chip must also handle bigger FPA (256 x 256), which require more internal memory within the chip. The windowing and staggered row FPA requirement will not effect the centroiding design since all proper control signals will be provided by the clustering chip automatically.

Also, two extra bits of resolution are needed out of the divider, so that overall resolution is down to 0.25 pixels. This will increase the signal to noise ratio that is "seen" by any subsequent object processing. Support for other features may be added as well. It remains to be seen if this function can be incorporated into the GT–VPSP.

3.4.4. GT–VDIT (Dithering)

The Georgia Tech VLSI Dither Processing (GT–VDIT) IC is used to support a mirror dithering method for clutter rejection. This method was developed by Lockheed Missile Space Corp. (LMSC) as a part of the LATS program.

The input pixel stream for dithering is broken into two time multiplexed phases: Staring and Scanning. During the staring phase, the focal plane is accumulating fixed frame information. The staring phase can be thought of as integrating energy from a target. During the scanning phase, the primary mirror is moved by piezoelectric actuators. This movement causes the input image to move in a small circular pattern on the focal plane. This motion is termed dithering due to the small deviation from center that is desired. The scanning phase can be thought of as integrating energy from the background. In a simplistic sense, if the scan energy (background) is subtracted from the stare energy (target), any background clutter is removed. The advantage of this approach is that both the target energy and the background energy are received by the same physical detector element. This significantly reduces the fixed pattern noise associated with staring focal plane arrays.

GT–VDIT is designed to accept the time multiplexed input data, along with actuator synchronizing signals and produce an output that effectively removes background clutter. The mathematical representation at each pixel position for dither processing is,

$$P_{out}(n) = \frac{1}{t_{stare}} \sum_{m=k+l}^{p} P_{in}(m) - \frac{0.5}{t_{scan}} \left[\sum_{m=1}^{k} P_{in}(m) + \sum_{m=p+q}^{r} P_{in}(m) \right], \qquad [1]$$

subject to the time line in Figure 13.

Figure 13. Dithering Operation Time

Notice that this IC also decimates the input. From the time line, the decimation appears to be r. The actual decimation is p because the second scan for pixel n is used as the first scan for pixel n+1. The goal input frame rate for GT–VDIT is 1000 frames per second. The IC is being designed with provisions to allow the use of FPAs with larger size and/or higher frame rates. Windowing provisions are also being incorporated into GT–VDIT.

3.4.5. GT–VOFX (Object Feature Extraction)

The goal behind the Georgia Tech VLSI Feature Extraction (GT–VOFX) chip set is to provide target information to the Executive Processor. This additional information will enhance the Executive Processor's ability to distinguish between different targets. The proposed Object Feature Extraction chip set is diagrammed in Figure 14. The pixel information coming from the FPA would first be processed by GT–NUC, GT–VTF, GT–VSF, and GT–VTHR chips. The output of the GT–VTHR chip would then be sent to the Object Feature Extraction pipeline. The first function to be performed is a windowing of the image frame around a target of interest. The coordinates of the centroid to the target of interest are provided by the Executive Processor. This window is then processed to extract image feature information. This information, which is unique to a specific target, is sent to a Neural Network, which will serve to associate the feature set with a target. The output of the Neural Network is sent to the Executive Processor. The Executive Processor can then combine this information with other information it has gathered to make determinations about the target of interest.

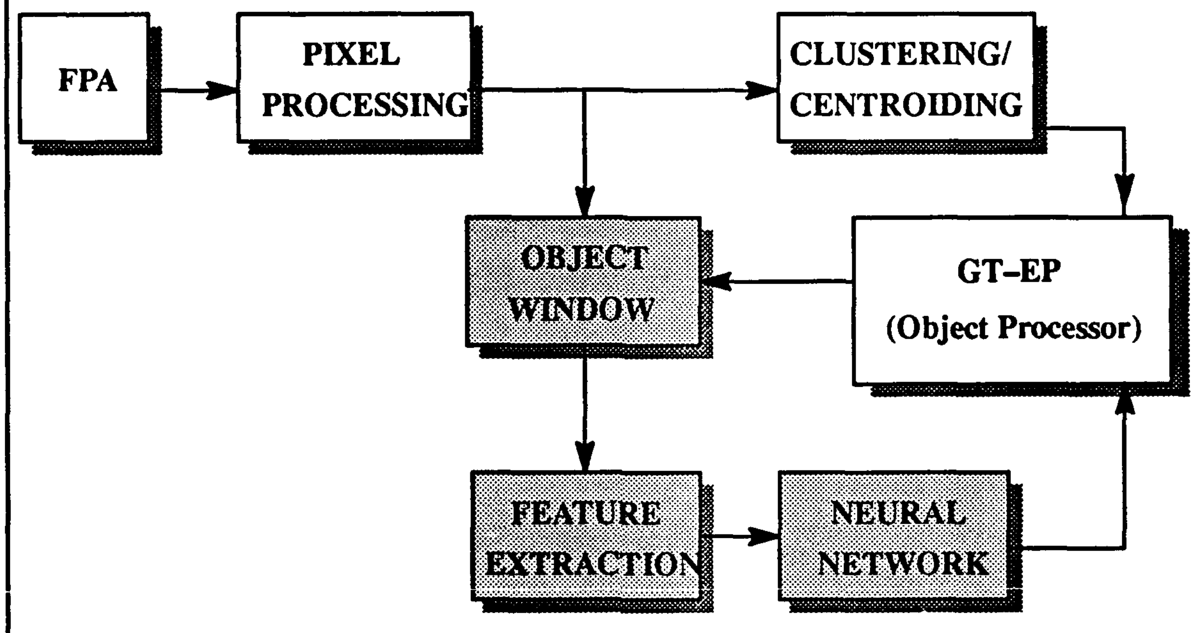


Figure 14. Object Feature Extraction

The Object Window allows the selection of a specific target within the image when several targets are present. This, in turn, reduces the complexity of the subsequent chips by reducing the amount of information that must be processed. Within the area of windowing, there are issues to be addressed which are unique to the Feature Extraction. The main issue is that of having partial targets in the window along with the target of interest. A second issue concerns the case of when the target is too large to be completely visible within the window. The steps to be taken in the development the Object Window chip are: determination of the window size to use for optimal performance; development of an algorithm to perform the actual windowing; the development of algorithms to address the issue of multiple targets and targets to large to fit within the window; the establishment of control signals for the coordination of windowing with the video pipeline.

The Feature Extraction chip will provide a set of parameters which are unique to a given target and which are invariant to scale and rotation. The current research is centered around calculating the moments of the target and algebraically combining them to form invariant moments.[35]

This approach is an extension of the the function carried out by the Centroiding Chip. Another choice would be to use Fourier Descriptors to categorize the target[36]. The invariant moments are sensitive to the intensity profile while being less sensitive to shape distortions. The Fourier Descriptors are the opposite. They are more sensitive to shape distortions and less sensitive to the intensity profile. It maybe that the final analysis will show that a combination of both types of features will yield the optimal solution. The development of the Feature Extraction chip will be closely tied to the development of the Neural Network chip. The first step is to determine what set of features combined with the Neural Network will achieve the best results. Once a set of features have been selected, the hardware implementation will need to be developed including all control and synchronization schemes.

The Neural Network will serve the purpose of associating the feature parameters with a target. The Neural Network currently under investigation is shown in Figure 15. Figure 15 shows the full interconnection for one node in each layer. It does not show all the interconnections nor the connections for learning. This is for the sake of legibility. It employs the Back Propagation model with full interconnection between layers[37]. The Back Propagation model consist of an input layer of fanout nodes. These nodes perform no computations but are elements which serve only to distribute the input to the first hidden layer. The input layer is followed by one or more hidden layers of processing elements. The current design only incorporates one hidden layer. These hidden layers consist of many processing elements which receive weighted inputs from the previous layer. These processing elements sum their inputs and then apply a non–linear transfer function to the total. This result is serves as input to the next layer after being multiplied by a weight specific to that connection. Following the hidden layer(s) is an output layer. This layer functions the same as a hidden layer with the exception that it's output serves as the circuit output. The main advantage of the Neural Network is that it is able, through training, to associate an output with a given set of inputs without a precise mathematical relationship being known apriori. Through training, the circuit can be taught to distinguish between different inputs. The closer together the different inputs, general the more complex the neural network must be to distinguish between the two. The training requires the presentation of a set of input parameters associated with a target and the desired output. The output generated by the network is compared to the desired output and an error term is generated. This error term is back propagated through the network with each layer adjusting it's weighting of it's inputs so that the net result has the network output moving closer to the desired output. This is repeated for all targets of interest until the neural network has achieved a desired level of operation. The learning process need not be performed on-line. A simulator may be used to determine the weights to be used and then these weights can be loaded into the circuit. An advantage of a Neural Network is it's ability to determine relationships among the input data that are not readily obvious. Also a Neural Network is not unique to a specific set of targets. To classify a new set of targets only requires a new set of weights be loaded into the current hardware. This flexibility allows the Neural Network to be quickly reconfigured to recognize a new set of targets or to modify the targets it currently recognizes. The Back Propagation model has the advantage of being a feedforward network with communication required between layers but not between nodes on a layer. This reduces the communication overhead substantially. Also, each output gives an indication of the probability of a target being in that class. This is opposed to other models which only determine which classification the target is closest to while providing no information on how close to other classifications it is. Most neural network implementation are analog. The desire for the Object Feature Extraction chip set is to use a digital neural network. A digital network will allow greater accuracy[37] than obtainable through an analog implementation but at the sacrifice of chip area. A digital implementation will allow easier development of time multiplexing of the processing elements should this need arise as compared to an analog version. The development steps for the neural network are: review current models available and determine the one best suited for this problem; determine the minimum size network needed for implementation; develop a digital processing element; determine if the network can be fully implemented or will multiplexing of elements be necessary; develop the control signals for interface with the system.

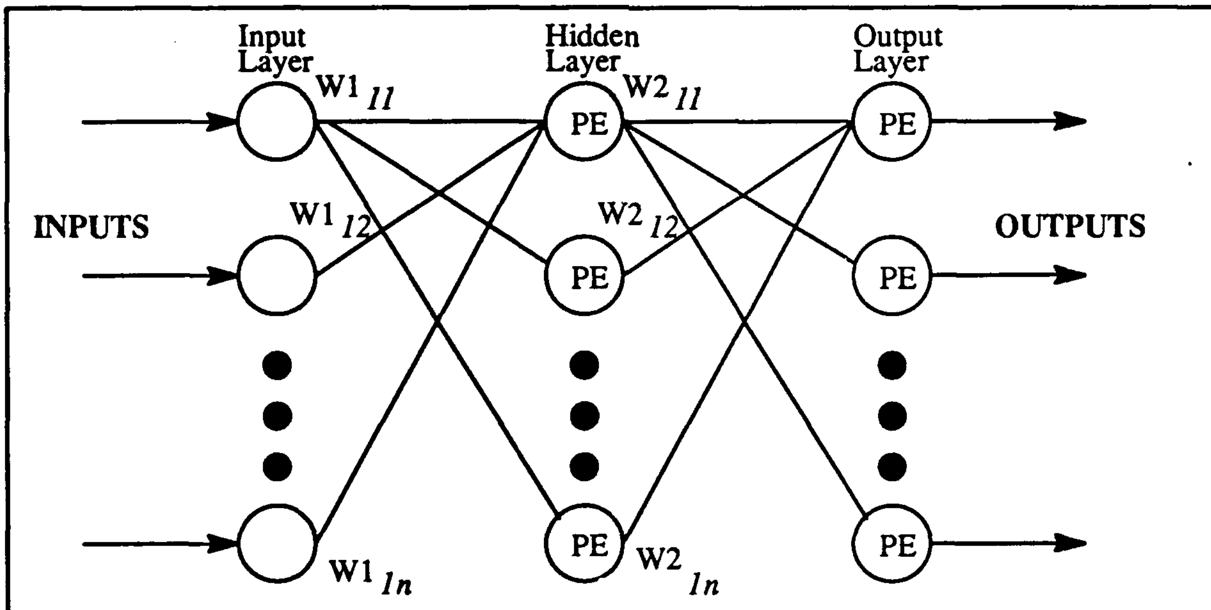


Figure 15. Neural Network

3.4.6. GT–VSEF (Sensor Fusion)

Many future seeker designs involve the use of multiple sensors. Typically these involve multiple FPA's operating over a wide range of frequencies (e.g. long–wave IR up to visible). More exotic technologies (such as laser radar or "lidar") may be incorporated as well. Once the information from individual sensors has been processed down to a manageable amount of data, it will be necessary to combine information from these several source in an intelligent manner. The purpose of using multiple sensors is so that information of fundamentally different natures can be combined. One example of this is temperature estimation, which requires intensity measurements from at least three different frequency bands. Lidar may provide some distance–to–target information as well. It is not immediately clear what information must be combined, but it is clear that this function will have to be performed on a real–time basis.

There are two candidates for this function. The first would be some sort of arithmetic computing engine which would combine numerical data using some pre–defined algorithm. That is, something like an Executive Processor may do the job. A more ambitious program would be to use Neural Network technology to combine the information in some meaningful way. This would be especially appropriate if many fundamentally different sensors were to be fused.

Further study into which sensors to support is needed. The research on neural net is described in Section 3.4.5.

3.4.7. GT–VPSP (Programmable Signal Processor)

The main goal of GT–VPSP effort is to have one chip design that can be programmed (by software programming) to perform all the algorithms needed for pixel processing, which include non–uniformity compensation, temporal filtering, spatial filtering, and thresholding (see NO TAG). In addition, it must be able to handle other enhancements such as bigger FPA size (256 x 256), variable frame rate by windowing, and staggered row FPA.

The type of architecture that will be implemented is being investigated. It must handle 100 Hz frame rate for 128 x 128 frame window. The design of the datapath, memory interface, and control unit are mainly driven by this real time performance requirement. To increase signal–to–noise ratio and to simplify arithmetic overflow handling, the internal number format to use would be 22–bit floating point number (1–bit sign, 5–bit exponent, and 16–bit mantissa).

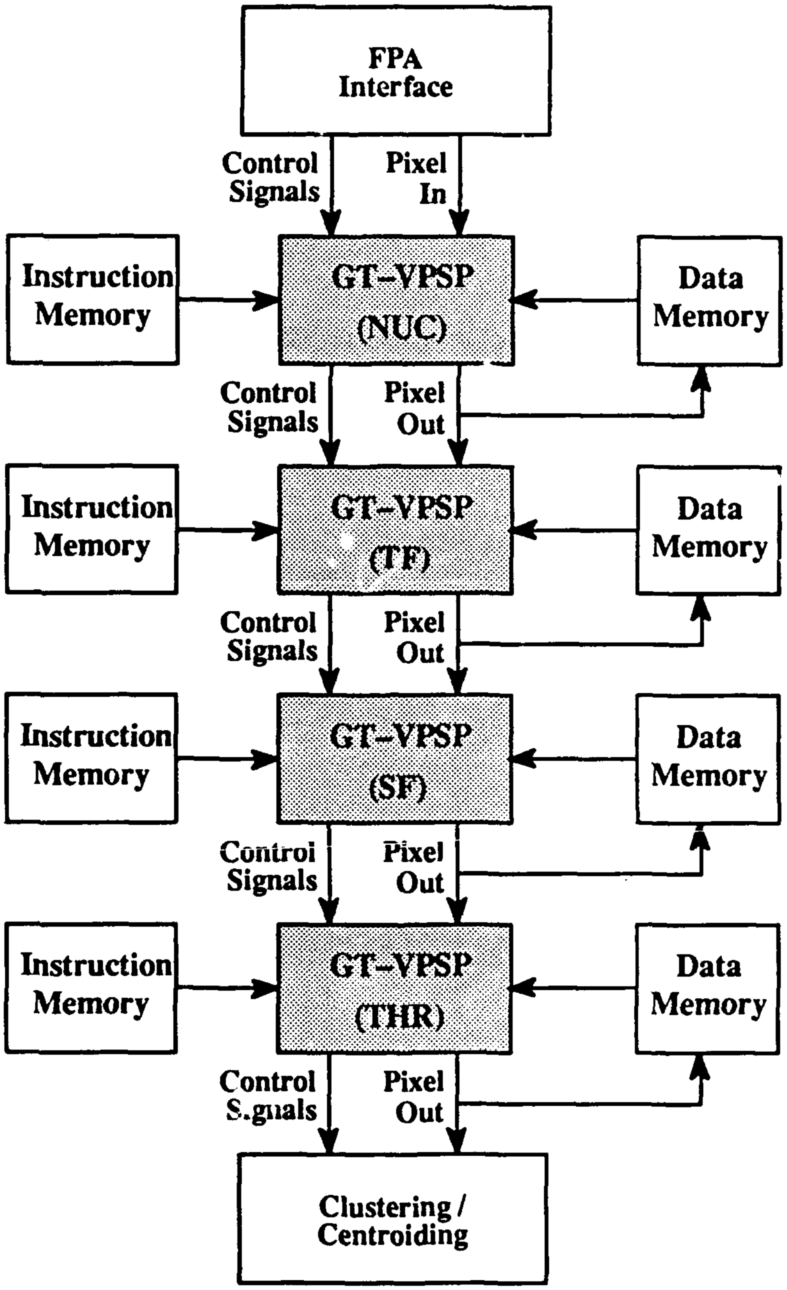


Figure 16. Pixel Processing Using GT–VPSP

4. Summary

Georgia Tech has been developing a set of modular VLSI chips that can be used to construct a light–weight, low–power, high performance flight computer to guide, navigate, and control (GN&C) advanced kinetic energy weapon (KEW) interceptors. This VLSI development effort has led to two complete processsors (the data processor GT–DP and the executive processor GT–EP) where six VLSI chip designs (GT–VSEQ, GT–VDR, GT–VSNI, GT–VFPU, GT–VIAG, and GT–VDAG) were successfully fabricated, tested, and integrated as a system. In the third processor (signal processor GT–SP), four VLSI chips (GT–VSF, GT–VTHR, GT–VCLS, GT–VCTR) were successfully designed, fabricated, tested, and integrated as a system. The remaining two chips (GT–VNUC and GT–VTF) are currently in fabrication. These three processors can meet the stringent real–time processing requirements of high performance, complex GN&C systems.

A total of 12 chips had been designed and developed. All except the GT–VTF and the GT–VNUC chips had been fabricated and tested. All the chips tested had functioned as expected. The problems encountered so far had been minor. When given an improper sequence of frame control sequence, the GT–VSF must be turned off to continue function properly. Furthermore, the reading of a centroid message from the GT–VCLS chips must be done without an interruptions in between. Work -rounds are availbale for both of these problems, the former by making sure that the GT–VSF does not receive an improper sequence of frame control signals, the later by disabling the interrupts while reading the centroid data.

An inevitable consequence of this successful result is the desire to improve on working designs once areas of improvements become apparent. Georgia Tech is challenged by new additional requirements and other possibilities to enhance some designs. In the executive processor, the emerging requirement for double–precision floating point direct implementation is being pursued. The aggressive advancement in CMOS device scaling has also led Georgia Tech to integrate the existing multiple chips into one chip. To speed up communication among processors, the interconnection chips will support parallel data transfers with non–blocking mechanism.

Many new requirements have emerged since design work began on the signal processing hipset, such as: bigger frame size (256 x 256), variable frame rate by windowing, staggered row FPA, more precision on the centroids, and additional signal processing algorithms. To add flexibility and adaptability to various algorithms, a special purpose superscalar programmable signal processor chip is being developed. This single processor chip design must be able to perform the existing non–uniformity compensation, temporal filtering, spatial filtering and thresholding algorithms, and also meet and support the additional requirements.

Some additional signal processing algorithms have been identified as candidates for direct silicon implementation. These include dithering (which compensates for a scanning motion by the seeker), object feature extraction (which involves some neural networks algorithm), and sensor fusion. The latter would combine information from a variety of sensors (e.g. visible, near and far infrared and laser radar).

Finally, the integration of the GN&C processor based on the existing set of VLSI chips is now taking place. It is important to evaluate the GN&C processor in the context of the overall mission requirements. The integration of the GN&C processor with the Parallel Function Processor in a hardware–in–the–loop testing environment will provide many useful insights into the good and the bad aspects of the existing designs. These lessons must be used to lay the foundation for the next generation chip set.

A schedule of development is found in Figure 17.

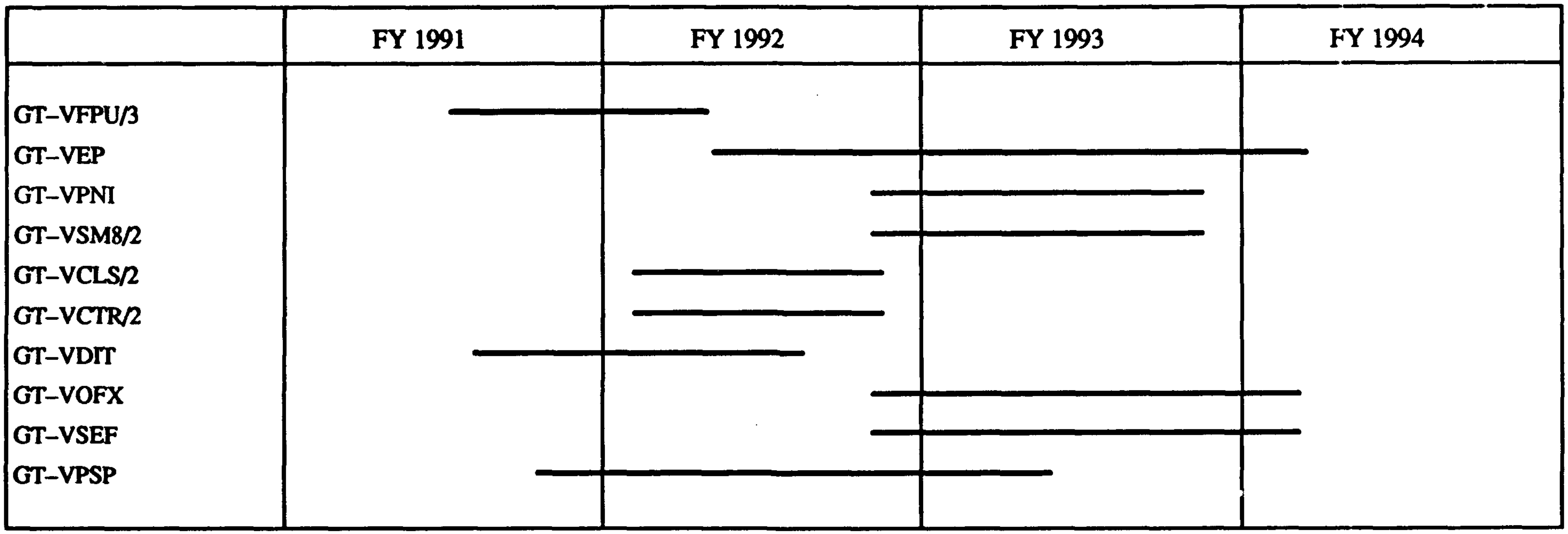


Figure 17: VLSI Development Schedule

5. References

[1] Wei Siong Tan, "Serial Network Interface Chip: GT–VSNI/1," VLSI Design Document, May 12, 1988, CERL008–0010.1

[2] Wei Siong Tan, "Serial Network Interface Chip: GT–VSNI/1," Design Verification Document, June 15, 1988, CERL008–0011.1

[3] Amar Ghori, "Eight Point Crossbar Switch Chip: GT–VSM8/1," VLSI Design Document, May 26, 1988, CERL008–0020.1

[4] Amar Ghori, "Eight Point Crossbar Switch Chip: GT–VSM8/1," Design Verification Document, May 26, 1988, CERL008–0021.1

[5] Sam Russ, "Floating/Fixed Point ALU Unit: GT–VFPA/1," VLSI Design Document, June 28, 1988, CERL008–0030.1

[6] Sam Russ, "Floating/Fixed Point ALU Unit: GT–VFPA/1," VLSI Design Document, August 17, 1988, CERL008–0030.2

[7] Sam Russ, "Floating/Fixed Point ALU Unit: GT–VFPA/1," Design Verification Document, August 17, 1988, CERL008–0031.1

[8] Sam Russ, "Floating/Fixed Point ALU Unit: GT–VFPA/1," Design Development Document, August 17, 1988, CERL008–0032.1

[9] Amar Ghori, "Sequencer: GT–VSEQ/1," VLSI Design Document, March 26, 1989, CERL008–0040.1

[10] Amar Ghori, "Sequencer: GT–VSEQ/1," Design Verification Document, March 26, 1989, CERL008–0041.1

[11] Sam Russ, "Instruction and Data Memory Unit: GT–VDR/1," VLSI Design Document, August 17, 1988, CERL008–0050.1

[12] Sam Russ, "Instruction and Data Memory Unit: GT–VDR/1," Design Verification Document, August 17, 1988, CERL008–0051.1

[13] Sam H. Russ, Toshiro Kubota, and Wei Siong Tan,"Data Address Generation Chip: GT–VDAG," VLSI Design Document, July 31, 1991, CERL008–0060.1

[14] Sam H. Russ, Toshiro Kubota, and Wei Siong Tan, "Data Address Generation Chip: GT–VDAG," Design Verification Document, July 31, 1991, CERL008–0061.1

[15] Wei Siong Tan, "Data Address Generation: GT–VDAG/1," Programming Mod-

el Document, November 5, 1988, CERL008–0062.1

[16] Wei Siong Tan and Samuel H. Russ, "Data Address Generation: GT–VDAG/1," Programming Model Document (Second Version), July 3, 1990, CERL008–0062.2

[17] Sam H. Russ and Wei Siong Tan,"Instruction Address Generation Chip: GT–VIAG," VLSI Design Document, July 31, 1991, CERL008–0070.1

[18] Sam H. Russ and Wei Siong Tan, "Instruction Address Generation Chip: GT–VIAG," Design Verification Document, July 31, 1991, CERL008–0071.1

[19] Wei Siong Tan and Samuel H. Russ, "Instruction Address Generation: GT–VIAG," Programming Model Document, January 3, 1991, CERL008–0072.1

[20] Wei Siong Tan and Samuel H. Russ, "Instruction Address Generation: GT–VIAG," Programming Model Document (Second Version), June 7, 1991, CERL008–0072.2

[21] Rafik Braham, "Thresholding Chip: GT–VTHR," Design Verification Document, February 14, 1991, CERL008–0081.1

[22] Amar Ghori, "Spatial Filtering Chip: GT–VSF," Design Verification Document, May 13, 1991, CERL008–0091.1

[23] Joseph I. Chamdani, "Clustering Chip: GT–VCLS," Design Verification Document, December 28, 1989, CERL008–0101.1

[24] Joseph I. Chamdani, "Centroiding Chip: GT–VCTR," Design Verification Document, October 17, 1989, CERL008–0111.1

[25] Randy Abler and Wei Siong Tan, "64–Point Fully Connected Switch: GT–VFCS/1," Design Verification Document, July 18, 1989, CERL008–0121.1

[26] Prem Pahlajrai and Toshiro Kubota, "Non–Uniformity Compensation Chip: GT–VNUC," VLSI Design Document, 1991, CERL008–0130.1

[27] Prem Pahlajrai and Toshiro Kubota, "Non–Uniformity Compensation Chip: GT–VNUC," Design Verification Document, 1991, CERL008–0131.1

[28] Toshiro Kubota and Wei Siong Tan, "Non–Uniformity Compensation Chip: GT–VNUC," User Guide Document, July 5, 1991, CERL008–0132.1

[29] Tsai Chi Chuang and Andy Register, "Temporal Filtering Chip: GT–VTF," VLSI Design Document, 1991, CERL008–0140.1

[30] Tsai Chi Chuang and Andy Register, "Temporal Filtering Chip: GT–VTF," Design Verification Document, July 31, 1991, CERL008–0141.1

[31] Tsai Chi Chuang, Andy Register, and Wei Siong Tan, "Temporal Filtering Chip: GT–VTF," User Guide Document, July 8, 1991, CERL008–0142.1

[32] Wei Siong Tan, "GT–Seeker/Scene Emulator to GT–GN&C Processor," Georgia Tech Interface Document, July 8, 1991, CERL008–0170.1

[33] Wei Siong Tan, Samuel H. Russ, Cecil O. Alford, "GT–EP: A Novel High–Performance Real–Time Architecture," International Symposium on Computer Architecture, May 27–30, 1991, Ontario, Canada.

[34] W. S. Tan, A. Register, J. Chamdani, R. Abler, S. Russ, P. Phalajrai, T. C. Huang, T. Kubota, "Task 4, 5, 6, & 7: Software Development and GN&C Processor Development," Final Report Vol. 4, CLIN 0006, November 7, 1990

[35] R. C. Gonalez nad P. Wintz, "Digital Image Processing, 2nd ed", Addison–Wesley Publishing Company, 1987, pp 419–423

[36] A. Reeves, R. J. Prokop, S. E. Andrews, and F. P. Kuhl, "Three–Dimensional Shape Analysis Using Moments and Fourier Descriptors", IEEE Transactions onPattern Analysis and Machine Intelligence, Vol. 10, No. 6, November 1988, pp 937–943.

[37] Robert Hecht–Nielson, "Neurocomputing", Addison–Weslay Publishing Company, 1990.