

2

FINAL REPORT

DTIC FILE COPY

VOLUME 4

**TASK 4, 5, 6 & 7: SOFTWARE DEVELOPMENT AND GN&C
PROCESSOR DEVELOPMENT**

CLIN 0006

November 2, 1990

MACROSTRUCTURE LOGIC ARRAYS

Contract No. DASG60-85-C-0041

Sponsored By

The United States Army Strategic Defense Command

COMPUTER ENGINEERING RESEARCH LABORATORY

Georgia Institute of Technology

Atlanta, Georgia 30332 - 0540

Contract Data Requirements List Item F006

Period Covered: 1985-1990

Type Report: Final

DISTRIBUTION STATEMENT A

Approved for public release
Distribution Unlimited

90 11 9 010

DTIC
SELECTE
NOV 13 1990
S
7 CG D

AD-A228 565

DISCLAIMER

DISCLAIMER STATEMENT - 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) DISTRIBUTION STATEMENT - Approved for public release; distribution is unlimited.
- (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.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a REPORT SECURITY CLASSIFICATION Unclassified			1b RESTRICTIVE MARKINGS			
2a SECURITY CLASSIFICATION AUTHORITY			3 DISTRIBUTION/AVAILABILITY OF REPORT 1) Approved for public release; distribution is unlimited. 2) (continued on reverse side)			
2b DECLASSIFICATION/DOWNGRADING SCHEDULE			5 MONITORING ORGANIZATION REPORT NUMBER(S)			
4 PERFORMING 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 DASG60-85-C-0041		
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) Macrostructure Logic Arrays Volume 4: Tasks 4,5,6 & 7						
12 PERSONAL AUTHOR(S) C. O. Alford						
13a TYPE OF REPORT Final		13b TIME COVERED FROM 6/28/85 TO 11/2/90		14 DATE OF REPORT (Year, Month, Day) November 2, 1990		15 PAGE COUNT 35
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) Volume 4: Task 4, 5, 6, & Software Development and GN&C Processor Development 1. Introduction 2. GN&C Processor Hardware 2.1 Data Processor: GT-DP 2.2 Executive Processor: GT-EP 2.3 Signal Processor: GT-SP 2.4 Interconnection Network 2.5 Operative and Physical Characteristics 2.6 Technology Progression 3. GN&C Processor Software 3.1 Support Software 3.2 Flight Software						
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

Security Classification of this page

Distribution statement continued

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.



Accession No.	
NTIS GRA&I	✓
DTIC TAB	
Unannounced	
Justification	
By _____	
Date/Initial	
Author's Name	
Dist	REPORT NUMBER
A-1	

Security Classification of this page

FINAL REPORT
VOLUME 4
TASK 4, 5, 6 & 7: SOFTWARE DEVELOPMENT AND GN&C
PROCESSOR DEVELOPMENT

CLIN 0006

November 7, 1990

Authors

**Wei Siong Tan, Andy Register, Joseph Chamdani, Randy
Abler, Sam Russ, Prem Pahlajrai, Tsai Chi Huang, Toshiro
Kubota**

COMPUTER ENGINEERING RESEARCH LABORATORY

Georgia Institute of Technology

Atlanta, Georgia 30332 - 0540

Eugene L. Sanders

USASDC

Contract Monitor

Cecil O. Alford

Georgia Tech

Project Director

Copyright 1990
Georgia Tech Research Corporation
Centennial Research Building
Atlanta, Georgia 30332

Table of Contents

1. Introduction	1
1.1. History	1
1.2. Objectives	1
1.3. Requirements	1
2. GN&C Processor Hardware	3
2.1. Data Processor: GT-DP	3
2.1.1. Sequencer (GT-VSEQ)	6
2.1.2. Dataram (GT-VDR)	6
2.1.3. Fixed/Floating Point Unit (GT-VFPU)	6
2.1.4. Serial Network Interface (GT-VSNI)	6
2.2. Executive Processor: GT-EP	7
2.2.1. Instruction Address Generation (GT-VIAG)	7
2.2.2. Data Address Generation (GT-VDAG)	9
2.3. Signal Processor: GT-SP	9
2.3.1. Gamma Suppression (GT-VGS)	9
2.3.2. Non-Uniformity Compensation (GT-VNUC)	10
2.3.3. Temporal Filtering (GT-VTF)	10
2.3.4. Spatial Filtering (GT-VSF)	14
2.3.5. Thresholding (GT-VTHR)	14
2.3.6. Clustering (GT-VCLS)	14
2.3.7. Centroiding (GT-VCTR)	14
2.4. Interconnection Network	19
2.4.1. GT-VSM8	19
2.5. Operating and Physical Characteristics	19
2.6. Technology Progression	25
3. GN&C Processor Software	30
3.1. Support Software	30
3.1.1. Compilers	30
3.1.2. Utilities	30
3.2. Flight Software	30

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, and implementation of the algorithms in VLSI chips.

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.

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, thresholding, clustering and centroiding. 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.

All software for the GN&C flight computer is required to be programmed in Ada. As a result, the development of an Ada compiler for the GN&C processor is essential. A representative set of flight software algorithms is required for a functional verification of the GN&C processor. Basic software utilities are needed program loading and for the processor to host communication with the host platform. Initially the program code is stored in RAMs. Eventually, the RAMs will be replaced with ROMs.

2. GN&C Processor Hardware

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: data processor (GT-DP), 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. The architecture of the GN&C processor and its capabilities are presented in the following sections.

2.1. 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 k bytes). The Georgia Tech Data Processor was designed to meet these requirements. Four GT-DP processors are shown in Figure 2. The number can be changed up or down to meet specific KEW requirements.

As shown in Figure 2, the GT-DP processor consists of four functional blocks: Instruction Control, Data Control, Arithmetic Control, and Communication Control. The Instruction Control Unit is mainly responsible for the generation of instruction addresses. It receives status flags from the Arithmetic Control Unit and appropriately determines the next instruction address. It facilitates branch-lookahead for efficient pipelined arithmetic instruction execution. The Instruction Control Unit is implemented in a VLSI chip designated GT-VSEQ.

In each computing cycle, the Data Control Unit supplies two operands to the Arithmetic Control Unit. 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. The Data Control Unit is implemented in a VLSI chip designated GT-VDR.

The Arithmetic Control unit is used to perform the actual data computation. 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 Arithmetic Control Unit 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 in the Arithmetic Control Unit, and branch-look-ahead facility in the Instruction Control Unit. This feature enables the GT-

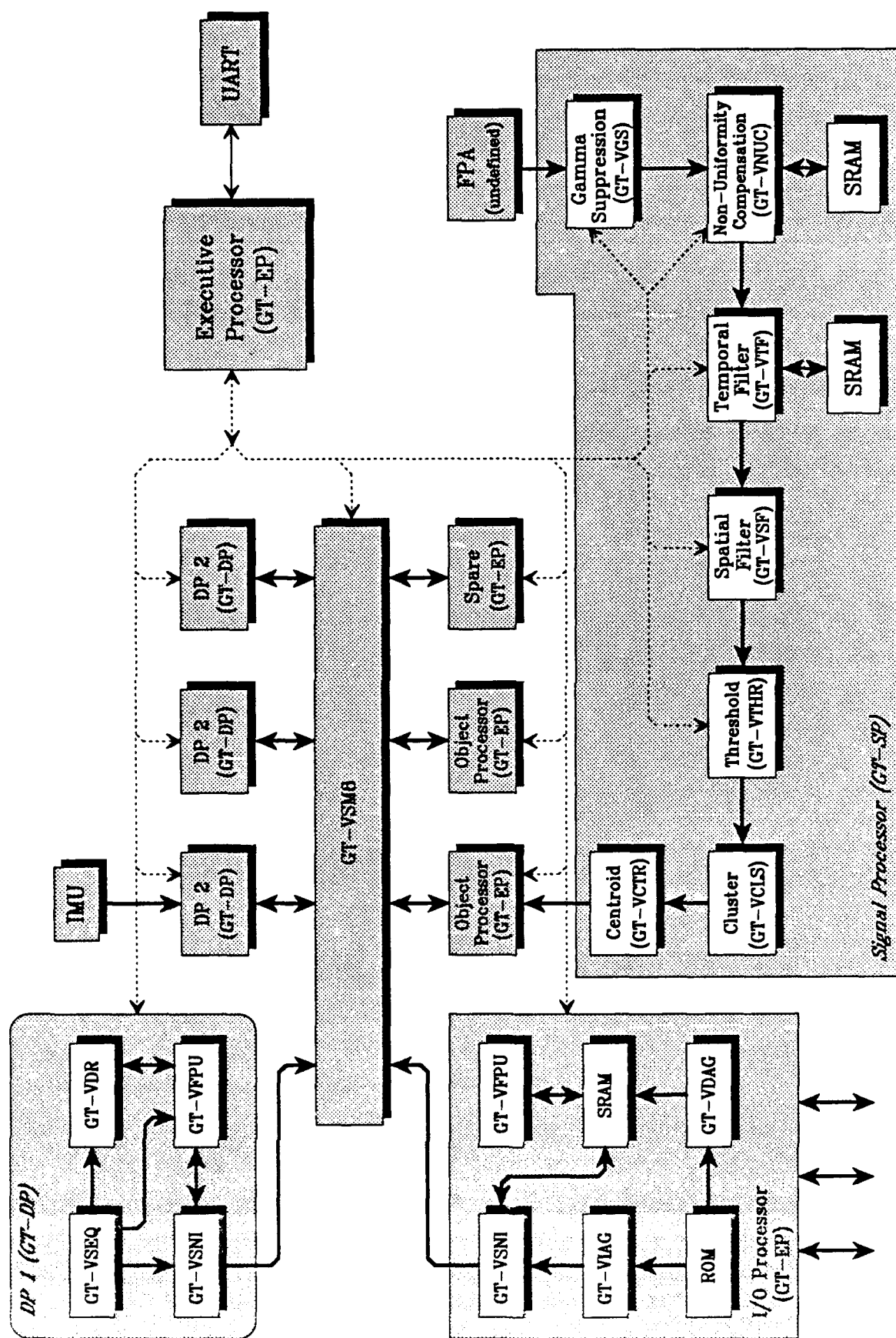


Figure 1. Architecture of the Georgia Tech GN&C Processor (GT-GN&C)

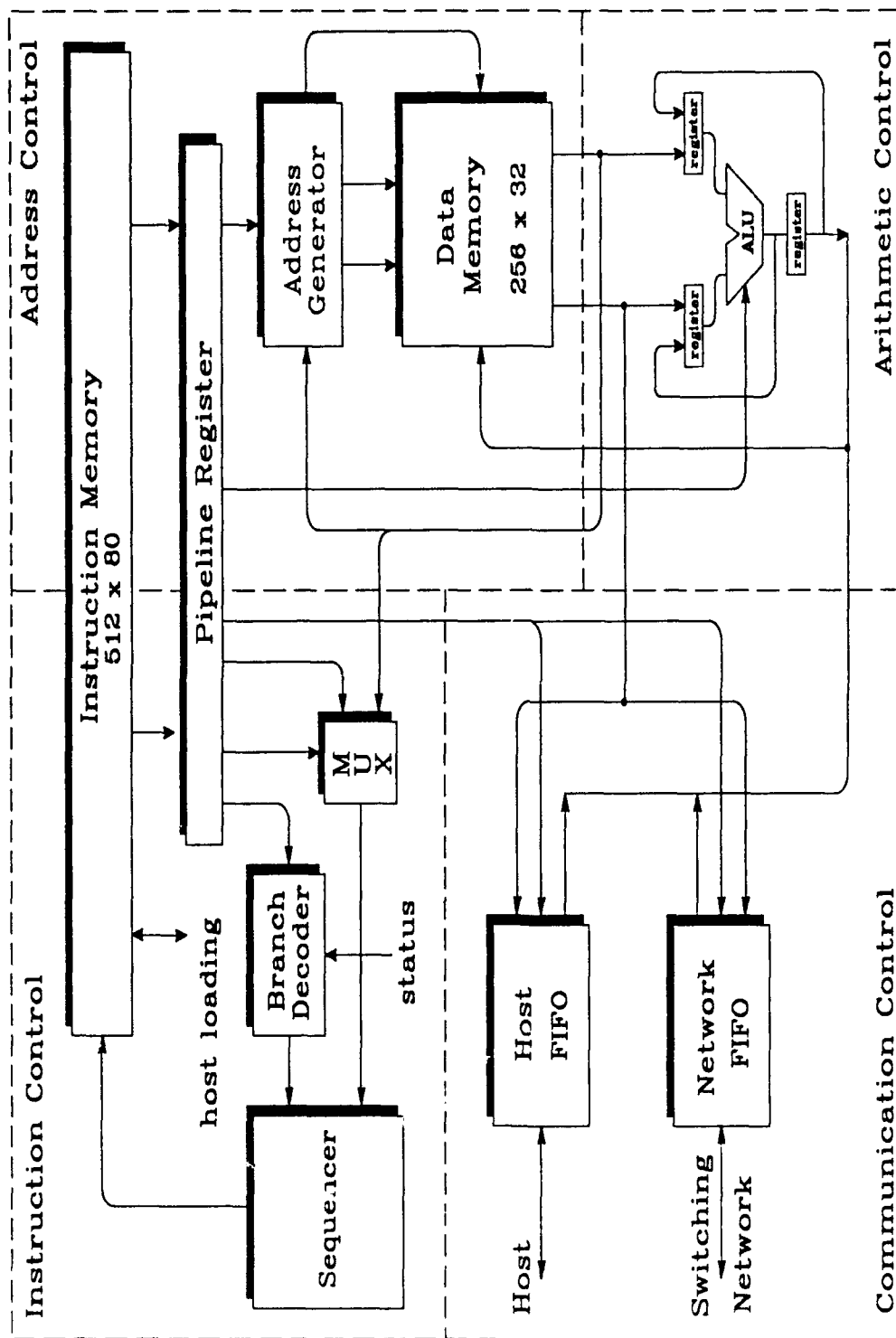


Figure 2. GT-DP Processor Architecture

DP processor to execute scalar computations efficiently. The Arithmetic Control Unit is implemented in a VLSI chip designated GT-VFPU.

The Communication Control Unit, designated as the GT-VSNI, is used to control the communication between the GT-DP 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.

2.1.1. Sequencer (GT-VSEQ)

Thirteen (13) fabricated parts have completed functional and speed tests. The chip is currently undergoing test at the processor module level. A complete memory test has been written for the Sequencer instruction memory. Conditional branching has also been tested. The design of the Sequencer chip is documented in [1] and [2]. The key parameters of the chip are listed in Figure 2.

2.1.2. Dataram (GT-VDR)

Only two working fabricated parts are available. Georgia Tech is asking Mentor Graphics to make a second fabrication run. Contractually, Mentor Graphics is obligated to deliver 5 working parts. The two working parts have been tested at the processor module level. The design is documented in [3] and [4]. The key parameters of the chip are listed in Figure 3.

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

Eight (8) fabricated parts have completed the functional test. They are currently being tested at the processor level. Some high level code has been successfully executed. A test board is currently being developed to allow the GT-VFPU to be tested at full speed. The design of the GT-VFPU chip is documented in [5] and [6]. The key parameters of the chip are listed in Figure 4.

2.1.4. Serial Network Interface (GT-VSNI)

Eight (8) fabricated parts have completed the functional test. One of the chips is used in a test setup for data communication between two GT-DP processor modules. The VLSI chip design is documented in [7] and [8].

2.1.5. The key parameters of the GT-DP chip-set are listed in Table 1.

Table 1. Key Parameters of GT-DP VLSI Chips

Chip	Die Size (milxmil)	Power (W)	No. Transistors (K)	Package	Technology
GT-VSEQ	371x410	0.7	131	100 PGA	NSC CMOS 1.5 u
GT-VDR	510x450	2.1	242	180 PGA	US2 CMOS 1.5 u
GT-VFPU	379x363	5.1	53	144 PGA	NSC CMOS 1.25 u
GT-VSNI	301x272	0.6	54	120 PGA	NSC CMOS 1.25 u

2.2. 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 3, 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 NO TAG). The network interface uses the GT-VSNI chip developed for the GT-DP processor. A more detail description of the GT-EP architecture is documented in [9].

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 prespecified 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.

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

GT-EP Processor Features :

- 32 Bit Data Word
- 25M SP Whetstones Performance
(Calculated based on 15M Whetstones Prototype)
- 16 Channels I/O Capability
- Expandable to
64M Words Data Memory
64M Words Instruction Memory
- 16 Real-time Interrupts
- Two 32-bit Built-in Timers
- Supports Ada Language
- Self-test Capability
- Supports Real-time Emulation and Control Capabilities
- Supports PFP & Flight Hardware

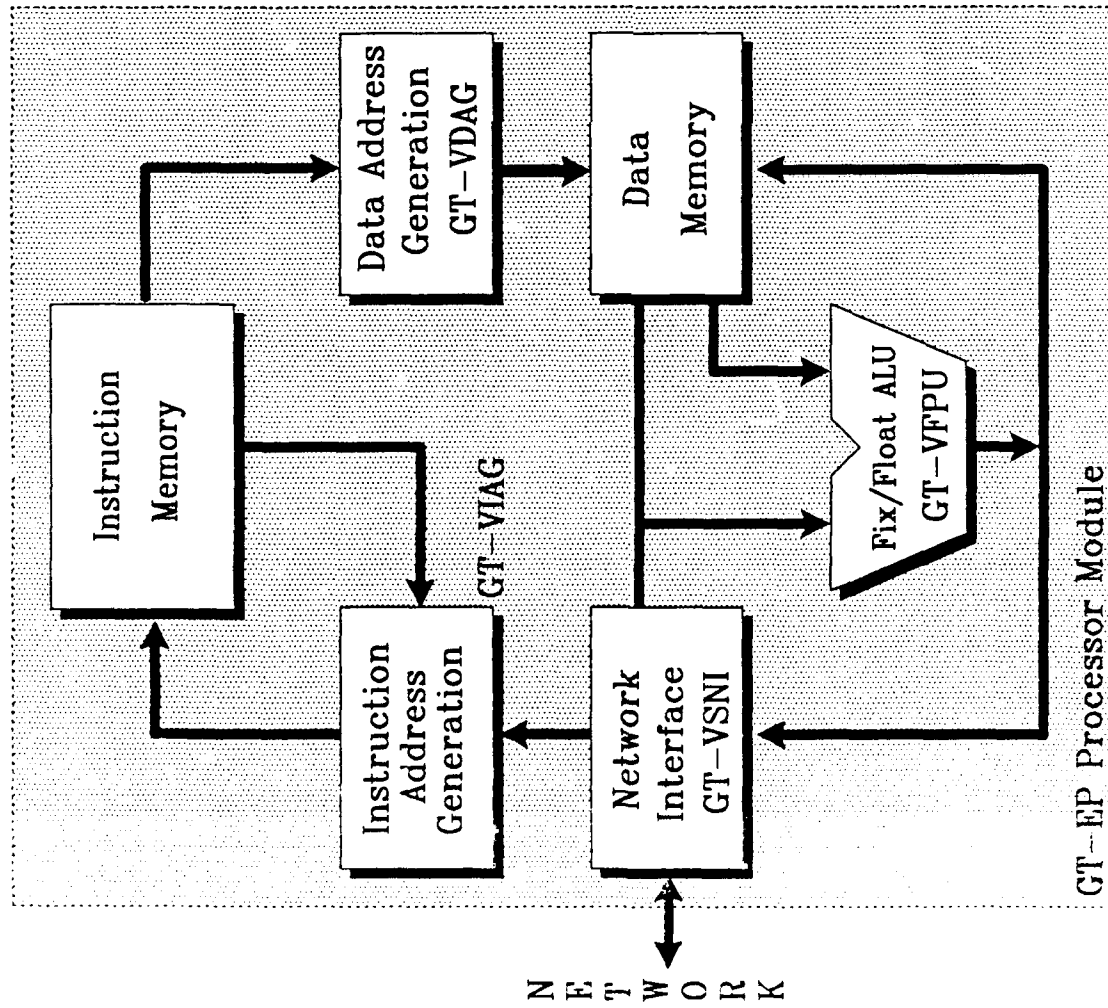


Figure 3. GT-EP Processor Architecture

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.

2.2.1. Instruction Address Generation (GT-VIAG)

The primary function of the GT-VIAG chip is to generate addresses for the instruction memory. In addition, it provides an opcode field to control signals to the GT-VFPU fixed/floating point arithmetic logic unit. For I/O functions, the GT-EP processor can directly access 16 input and output devices. Four of the 16 channels are reserved for asynchronous devices. The other 12 channels are used with synchronous devices. Each synchronous channel can be accessed in a single cycle, whereas the asynchronous channel requires a minimum of 6 cycles per access.

The GT-VIAG supports a static branch prediction scheme to reduce the branching time penalty associated with the pipelined GT-VFPU chip. A 32 level hardware stack is used for fast instruction address store and retrieve when executing procedure and interrupt calls. The GT-VIAG chip supports 16 vectored interrupts. Seven interrupts are used to handle exceptions for the GT-VIAG and GT-VDAG chips. Nine interrupts are available externally for general purpose, interrupt-driven, device interfaces. The GT-VIAG responds to all interrupts within 5 cycles. The interrupts are prioritized. Higher priority interrupts are honored while serving a lower priority interrupt. The GT-VIAG chip supports up to 16 MW of instruction address space.

A programming model of the GT-VIAG chip is documented in [10]. The functional test, floorplanning, and timing analysis have been completed. Manufacturing test vectors are being generated. Work on the VLSI design document and the design verification document is in progress. The design is expected to go out for design verification in February 1991.

2.2.2. Data Address Generation (GT-VDAG)

The GT-VDAG chip is used to generate two address fields for operand fetches and one address field for result store. The chip supports post-index addressing, for accessing arrays with constant strides, at a rate of one cycle per array element. Relative addressing is 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. The GT-VDAG chip supports a 64 MW data address space.

The GT-VDAG chip has completed all design stages. It is waiting for the completion of the GT-VIAG chip before going out for design verification and fabrication. The chip design is documented in [11], [12], and [13].

2.3. 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. The first functional block is the FPA interface (GT-VFPU) which is used to link the signal processor and the focal plane array. The FPA has not been designated. As a result this functional block is not defined.

2.3.1. Gamma Suppression (GT-VGS)

The second block is the GT-VGS which is used to address problems associated with gamma suppression. In order to filter out gamma spikes, the GT-VGS needs to process pixel data from the FPA at a very high frame rate (10,000 frames per second). Gamma spikes are represented by pixels that exceed a certain threshold. Pixels that fall below the gamma spike threshold are accumulated. Pixels that exceed the gamma spike threshold are suppressed (value set to zero). The threshold is calculated based on the current pixel value and a number of previous pixel values. This algorithm is illustrated in Figure 4. It has not been implemented in digital VLSI chips pending further study and development of gamma suppression schemes in the analog domain.

A program simulating several gamma suppression algorithms has been written. The simulation results agree with the results from the Lockheed LMSC simulation. The adjusted thresholding algorithm provides the best gamma rejection performance. This algorithm turns out to be very suitable for digital VLSI implementation. However, in order to prevent operating the D/A converters for the signal from the FPA at the high speed required by gamma suppression, Lockheed LMSC is proposing to implement the gamma suppression algorithm in analog. This approach reduces the D/A operating speed to approximately 100 frames per second from the 10,000 frames per second required by gamma suppression. The gamma suppression design effort has been placed on hold until the Lockheed analog effort is evaluated.

2.3.2. Non-Uniformity Compensation (GT-VNUC)

The third functional block, 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. During normal operation, each pixel value is mapped from one of the four linear segments to a common desired response. The functionality of the GT-VNUC is illustrated in Figure 5.

The control logic of the GT-VNUC is implemented in a VLSI chip. An external 1.3 Mb SRAM is required. The SRAM is required to be configured as 5 x 16k x 16. The design entry, functional simulation, floorplanning, and timing analysis for the GT-VNUC chip has been completed. Manufacturing test vectors are currently being generated. Work has begun on the preparation of a vlsi design document and a design verification document.

2.3.3. Temporal Filtering (GT-VTF)

The fourth functional block is the GT-VTF which performs time averaging of pixel values across frames. The GT-VTF is used to reduce random noise across frames as well as smearing of images due to

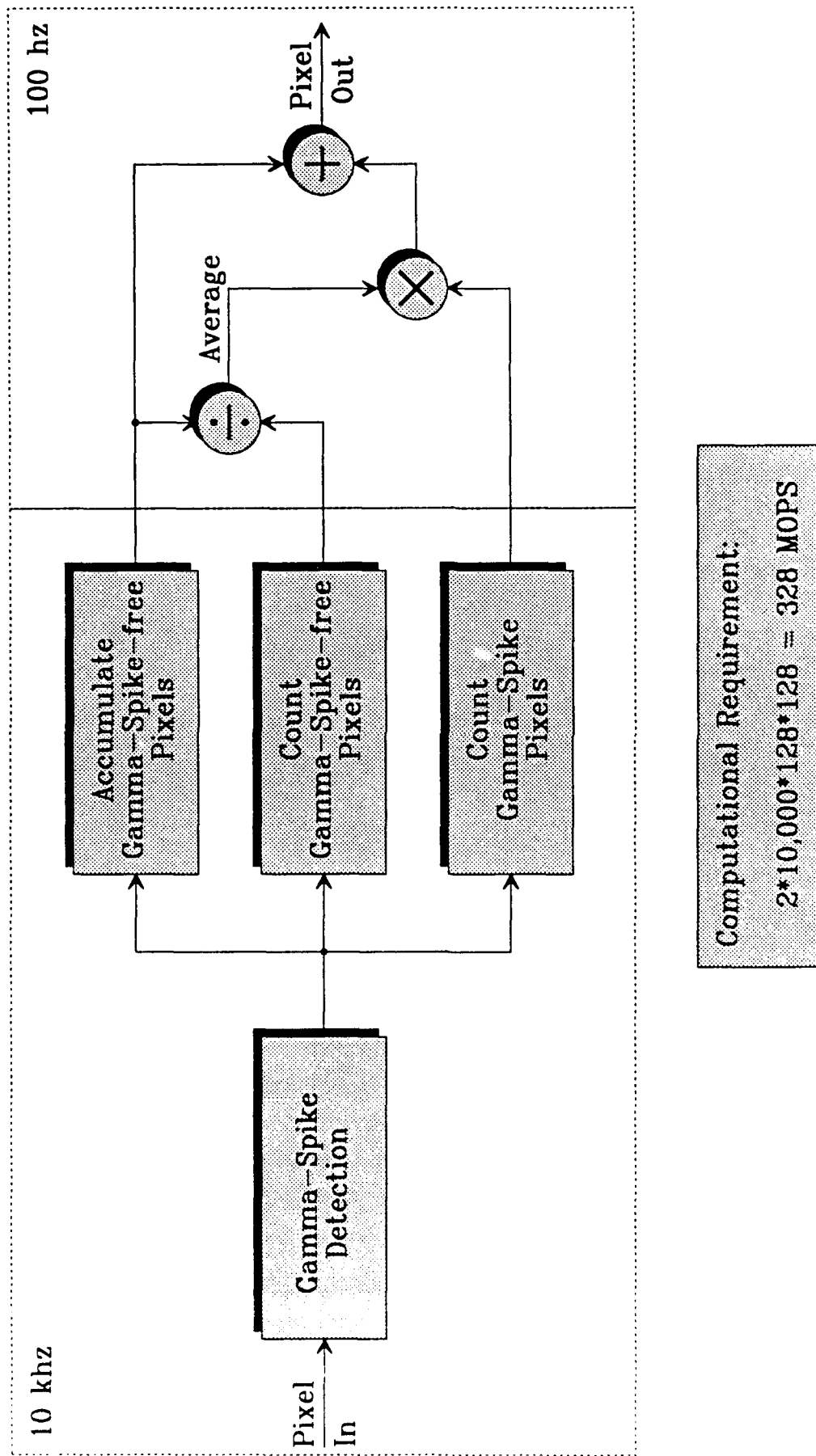
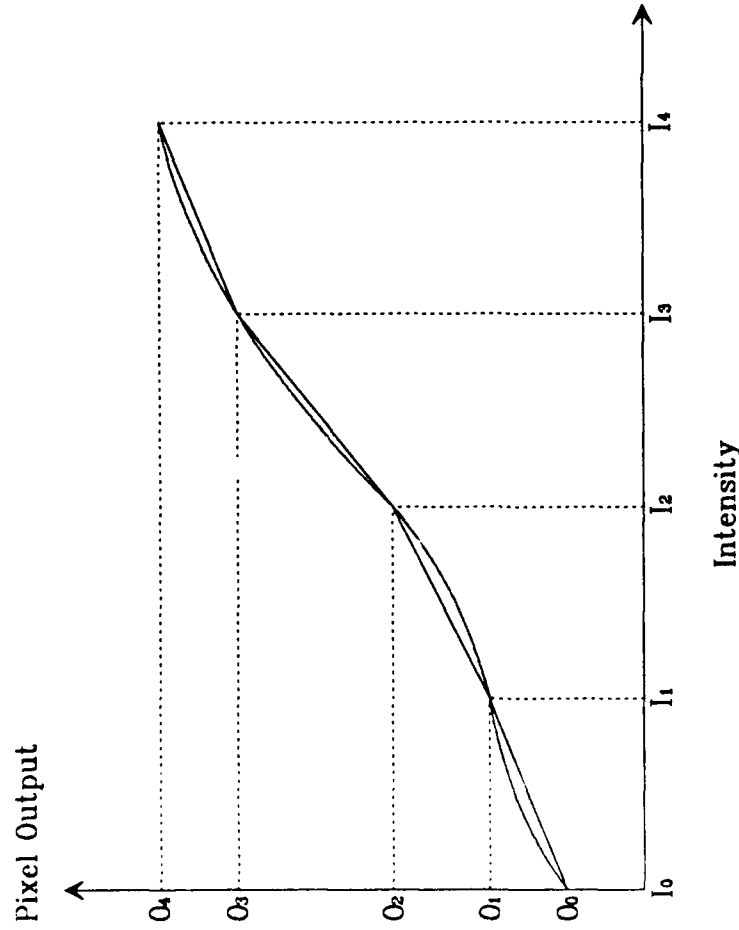


Figure 4. GT-VGS Gamma Suppression



- Storage Requirement $(5 \times 128 \times 128 \times 16) = 1.3$ Mbits
- Computational Requirement $(2 + 1 + 1 + 4) \times 128 \times 128 \times 100 = 13$ MOPS

For each pixel:

Calibration:

- Apply 5 known sources: $I_n, n = [0,4]$
- Sample Output responses: $O_n, n = [0,4]$
- Obtain 4 piecewise Linear functions:

$$PW(O_r, n) = \frac{(I_{n+1} - I_n)}{(O_{n+1} - O_n)} (O_r - O_n) + I_n$$

for $n = [0,3]$

Compensation :

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

Figure 5. GT-VNUC Non-Uniformity Compensation

jittering motion on the FPA. The GT-VTF is implemented as a fourth order temporal filter that makes use of pixel values from the previous four frames. Eight coefficients in the GT-VTF can be programmed from a host port to achieve a desired filter response. These coefficients can be dynamically changed to adapt to different operating environments. Figure 6 illustrates the filtering operation of the GT-VTF.

The functionality of the GT-VTF is implemented in a single VLSI chip. To store the previous four filter states, an external 64kx16 SRAM is required. The design entry, functional simulation, floorplanning, and timing analysis for the GT-VTF chip has been completed. Manufacturing test vector generation is in progress. The functional specification of the GT-VTF chip is documented in [14]. A VLSI design document and a design verification document are being generated.

2.3.4. Spatial Filtering (GT-VSF)

The fifth functional block is the GT-VSF, 9-point spatial filter. The GT-VSF performs filter operations based on the pixel value and that of its immediate eight surrounding pixels. The GT-VSF can be used to reduce the effects of spatial noise as well as to enhance/reduce the contrast of images. Figure 7 illustrates the functionality of the GT-VSF.

The GT-VSF VLSI chip design has been completed. Fabricated parts (55) have passed the functional test. The chip design is documented in [15] and [16]. The design is implemented in Hewlett Packard 1.0 micron bulk CMOS technology. the GT-VSF is currently waiting for insertion in the signal processor printed circuit board module.

2.3.5. Thresholding (GT-VTHR)

The sixth functional block, GT-VTHR, is used to suppress noise by cutting out pixels that exceed a constant or calculated threshold. Three types of thresholding are supported: simple, adjusted, and adaptive. Simple thresholding uses a constant lower threshold and fixed upper threshold set by the host. 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. The three thresholding modes of the GT-VTHR are illustrated in Figure 8.

The GT-VTHR chip design has been completed. The design package is ready for design verification at Mentor Graphics. The chip design is documented in [17] and [18].

2.3.6. Clustering (GT-VCLS)

The seventh functional block is GT-VCLS. As illustrated in Figure 9, 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 than 1. The diagonal distance between two pixels is considered a 1. Each cluster is tagged and merged with other clusters when two pixels, one from each cluster, touch.

The design of the GT-VCLS has been completed and fabricated. Fifty-four (54) tested parts are available. This chip is waiting for insertion into the signal processor module. The chip design is documented in [19] and [20].

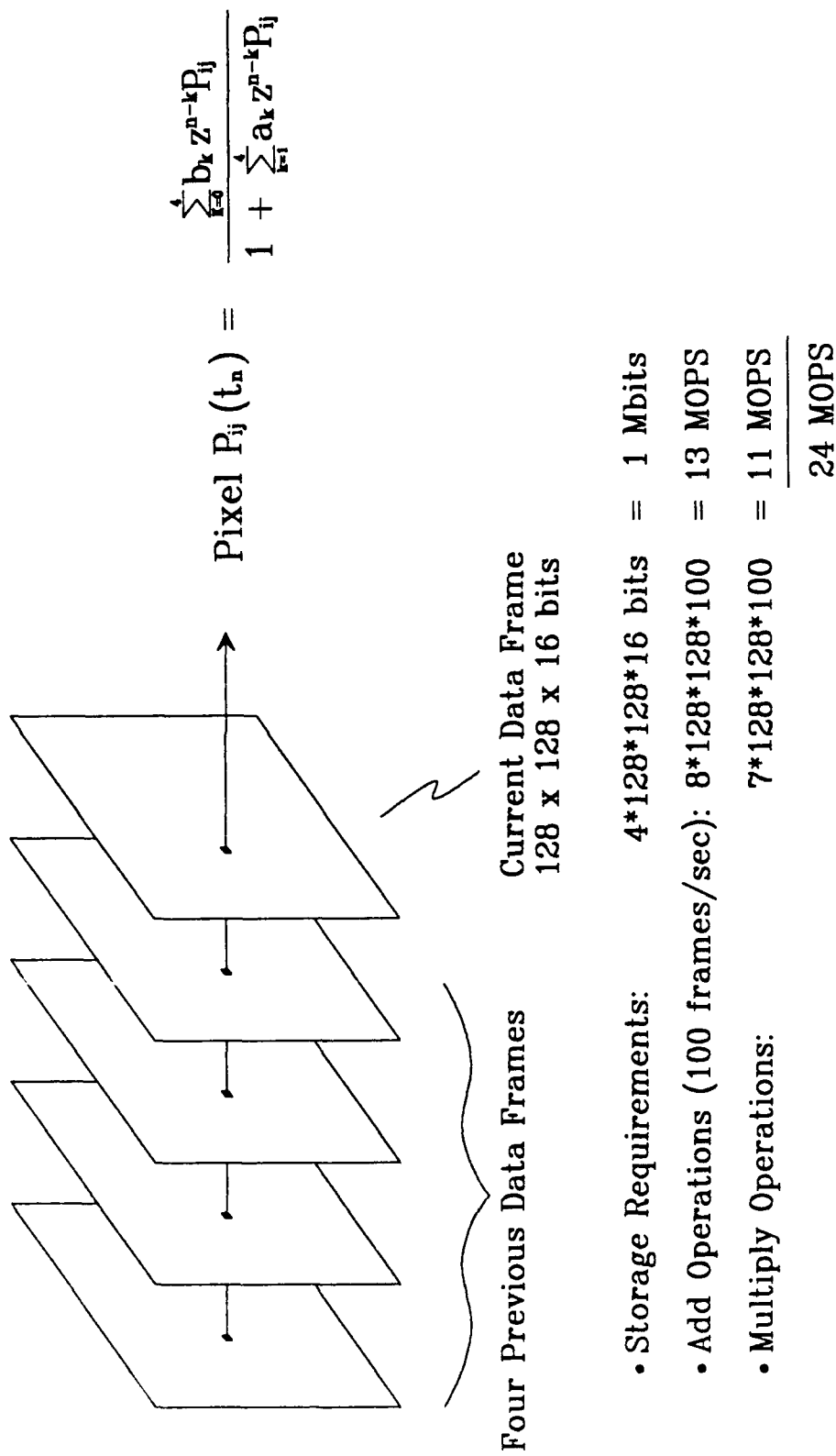


Figure 6. GT-VTF Temporal Filtering

9 Point, 2D, FIR Filter

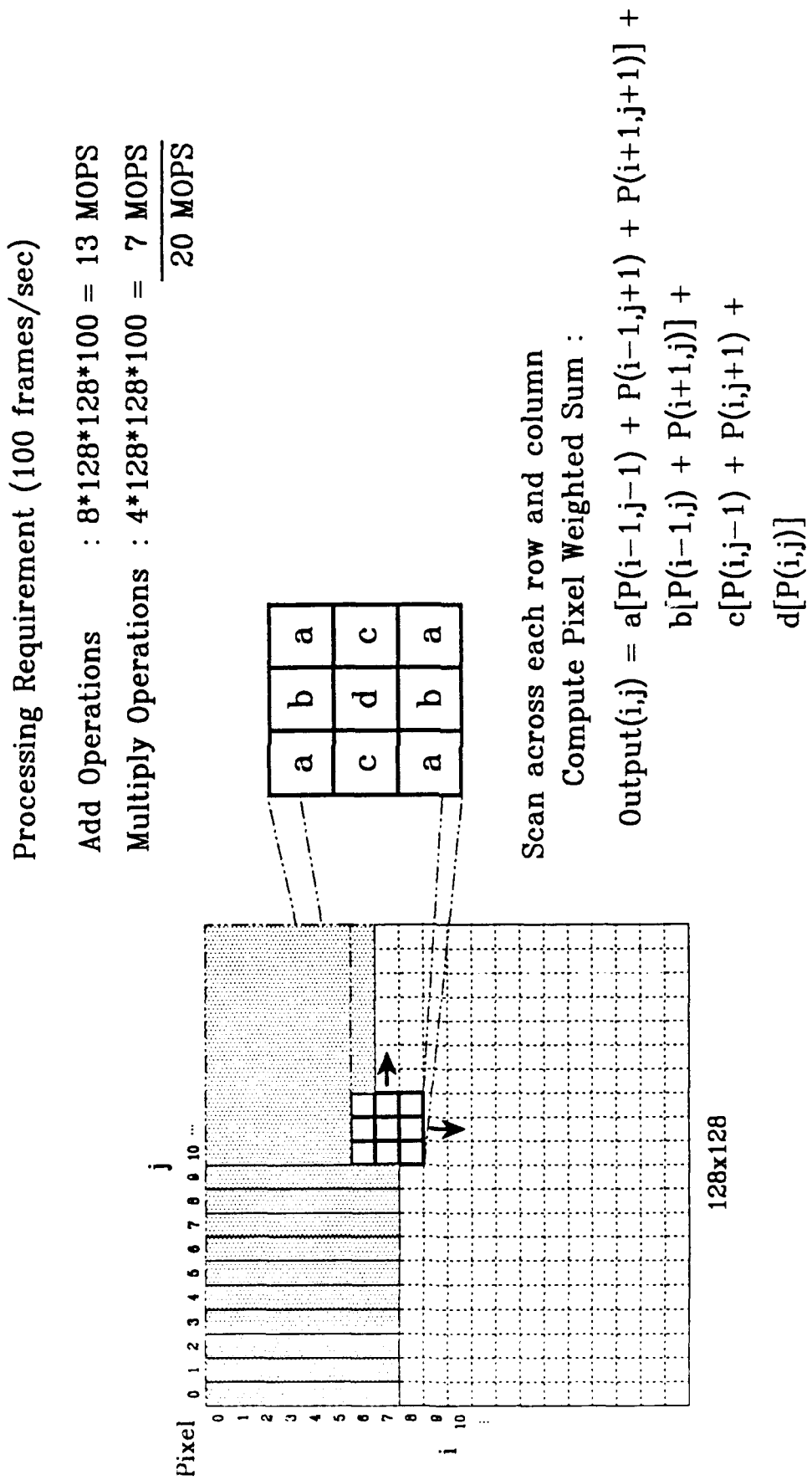
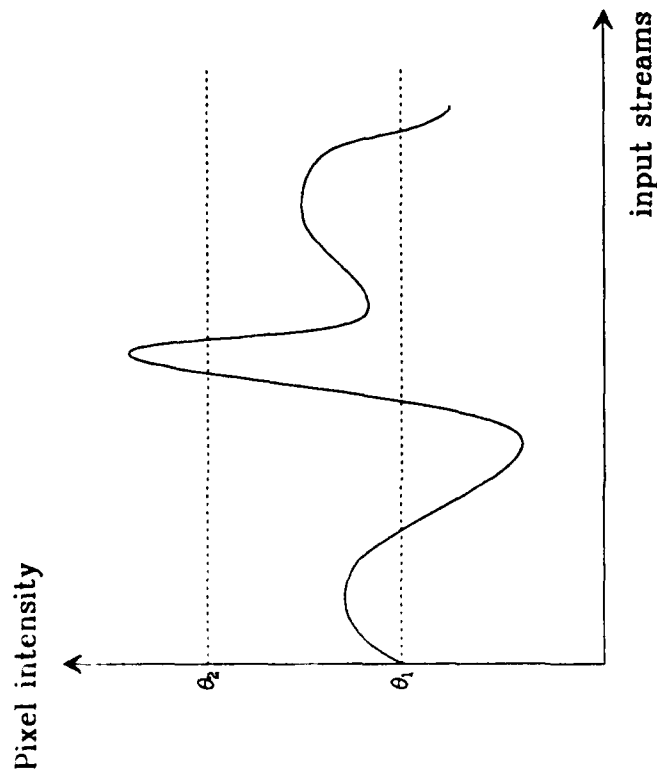


Figure 7. GT-VSF Spatial Filtering



3. Adaptive thresholding (53 MOPS)

θ_2 : constant

$\theta_1 = k_1 * E + k_2 L_1 + k_3$; $k_1 > 0$

E_1 = average value over neighboring 8 pixels ($8 * 128 * 128 * 100 = 13$ MOPS)

L_1 = absolute sum of the difference between E_1 and neighboring 8 pixels ($23 * 128 * 128 * 100 = 37$ MOPS)

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

1. Simple thresholding: θ_1 & θ_2 constants

2. Adjusted thresholding

θ_2 : constant

$\theta_1 = Z^{-1} \theta_1$ op delta

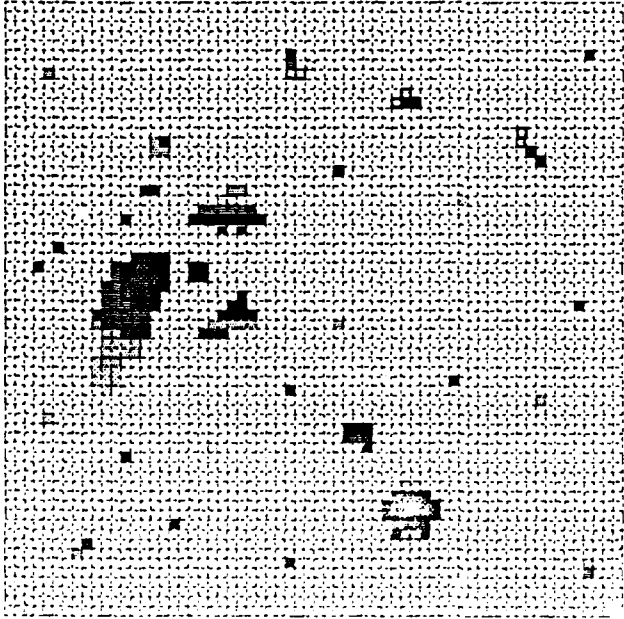
$$\text{op} = \begin{cases} +, & \# \text{ pixels} > N_1 \\ -, & \# \text{ pixels} < N_2 \\ \text{nop}, & N_1 < \# \text{ pixels} < N_2 \end{cases}$$

Z^{-1} : frame delay

pixels : total number of pixels in a frame

N_1, N_2 : constants controlled by the executive

Figure 8. GT-VTHR Thresholding



- Pixel $P(i,j)$ and $P(k,l)$ are adjacent if
 $(|i-k| < 2) \ \& \ (|j-l| < 2) \ \& \ [(j \neq k) \ || \ (j \neq l)]$
- Adjacent pixels with non-zero pixel intensity are grouped into a cluster
- Pixels are scanned from left to right and top to bottom
- A cluster is completed if no pixels in a row is adjacent to any pixels in the cluster
- Maximum number of clusters in a 128x128 FPA
is $64 \times 64 = 4096$
- Parallel algorithm for the identification and grouping of clusters requires 128 entries associative search
- Total equivalent processing requirement =
 493 MOPS

Figure 9. GT-VCLS Clustering

2.3.7. Centroiding (GT-VCTR)

The last functional block of the signal processor is the GT-VCTR centroiding chip. The statistical information of each cluster from GT-VCLS is computed by the GT-VCTR. As shown in Figure 10, the total intensity, the intensity centroid, the area (total number of pixels), and the area centroid of each cluster are calculated. Each finished cluster is sent to the object processor for target acquisition, tracking, and discrimination.

The design of the GT-VCTR has been completed. Fifty-five (55) fabricated parts have been functionally tested. The chip is waiting for insertion into the signal processor module. The chip design is documented in [21] and [22] .

2.3.8. GT-SP Chip Parameters

The key parameters of the GT-SP chip set are listed in Table 2.

Table 2. Characteristics of VLSI Chips

Function	Throughput (MOPS)	Power (Watts)	Size (Mil)	No. of Transistors	Design Status
GT-VNUC	12	1.2	391x382	60,000	Test vector generation
GT-VTF	23	0.9	421x426	76,000	Test vector generation
GT-VSF	18	0.8	378x401	40,000	Fabricated and tested
GT-VTHR	53	1.1	415x415	124,000	Design Verification
GT-VCLS	493	0.9	370x370	67,000	Fabriated and tested
GT-VCTR	50	1.1	395x395	117,000	Fabricated and tested

2.4. Interconnection Network

2.4.1. GT-VSM8

An 8-point fully-connected crossbar switch enables multiple processors to effectively exchange data and state variables (see Figure 1). The switching matrix chip, designated GT-VSM8, is designed to directly interface with the GT-VSNI chip. It provides 8 input ports and 8 output ports to connect up to 8 GT-VSNI chips. All processor modules communicate with the network through the GT-VSNI chip. The communication bandwidth from each processor is 40 Mb/s. Single bit error correction and double bit error detection is used for data communication through the GT-VSM8 chip.

The design of the GT-VSM8 chip has been completed. Eight (8) fabricated parts are available. The chip has been used to communicate data between two GT-DP processors. The chip design is documented in [23] and [24].

2.5. Operating and Physical Characteristics

Each GT-DP processor consists of 4 VLSI chips. The die size of each chip is shown in Figure 11. At an operating speed of 6.6 Mhz and a power consumption of 5.68 Watts, each GT-DP processor is capable

– For each cluster in the field of view calculate :

(A_x, A_y)
 (I_x, I_y)

Area : #A = total number of pixels in the cluster

Intensity : #I = Sum of all the intensity of the pixels in the cluster

Area Centroid : For all i and j such that pixels P(i,j) is an element of the cluster,

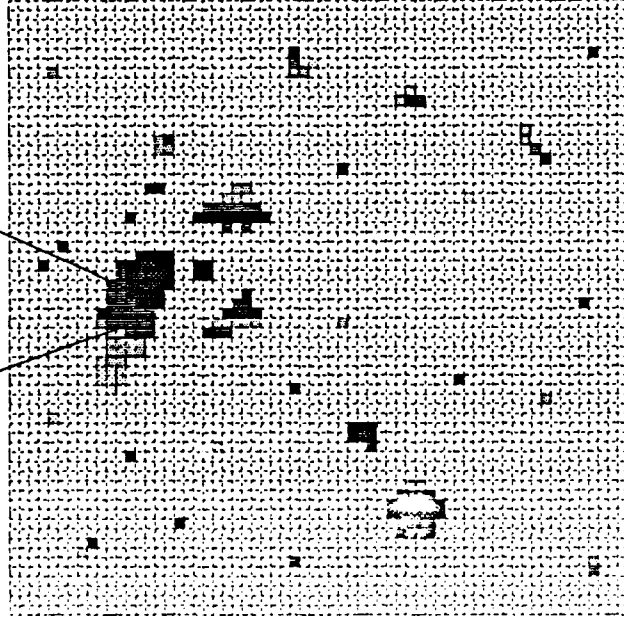
$$A_x = \frac{\sum x(i,j)}{\#A}$$

$$A_y = \frac{\sum y(i,j)}{\#A}$$

Intensity Centroid : For all i and j such that pixels P(i,j) is an element of the cluster,

$$I_x = \frac{\sum I(i,j) x(i,j)}{\#I}$$

$$I_y = \frac{\sum I(i,j) y(i,j)}{\#I}$$



Computational requirement :

25 MOPS

Figure 10. GT-VCTR Centroiding

Assumption : Each hybrid board with VLSI dies has a weight density of 4 oz/sq.ft.

Chip	Size (Mil2)	Power (Watts)
GT-VFPU	362x383	2.5
GT-VDR	501x524	1.85
GT-VSEQ	373x423	0.718
GT-VSNI	276x301	0.617
Total	642,025	5.68
Package	1"x1"x0.2", 80 grams / GT-DP	
Performance	6.6 MFLOPS	80 Mbps

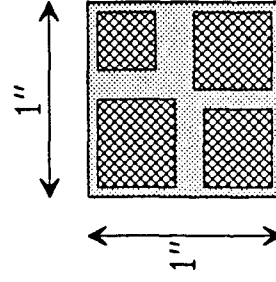


Figure 11. Characteristics of the GT-DP Processor

of executing 6.6 MFLOPS. Using General Electric High-Density Interconnect (HDI) hybrid packaging technology, the GT-DP processor can be packaged in a 1"x1"x0.2" space. Assuming that the hybrid board has a weight density of 4 ounces per square foot (equivalent to copper foil), each GT-DP processor would weigh approximately 80 grams.

The GT-DP processor contains two I/O ports. One of the I/O port is used for direct communication with the executive processor and has a bandwidth of 40 Megabits per second (Mbps). The second I/O port has a bandwidth of 40 Mbps and is used for parallel processing applications. The two I/O ports combined, provide an I/O bandwidth of 80 Mbps with a total of 36 I/O pins for the GT-DP processor.

The GT-EP executive processor consists of four VLSI chips (GT-VIAG, GT-VDAG, GT-VFPU, and GT-VSNI) and the necessary memory chips for the instruction and data memory. The number of chips required for the instruction memory depends on the amount of instruction and data memory used with the GT-EP processor. The width of the instruction memory (IW) is

$$IW = 32 + 3 \cdot \log_2 DM + \log_2 IM, \quad [1]$$

where DM is the number of words of data memory and IM is the number of words of instruction memory. The width of the data memory is 64 bits. In a typical configuration with 8k of instruction and data memory using off-the-shelf 8kx8 memory chips, the number of memory chips required is 8 for the data memory and 11 for the instruction memory. Assuming that each memory chip is 280x280 mils, using the GE HDI packaging technology, 8 memory dies can be placed on a single 1"x1"x0.2" hybrid circuit board. A second board can be used to hold another nine memory chips. The GT-VIAG and GT-VDAG VLSI chips are approximately 400x400 mils. The two chips and an additional memory die can be packaged on a third hybrid board. The total package size for the GT-EP executive processor with 8k of instruction memory and 8k of data memory will be 1"x1"x0.8". Each memory chip consumes about 0.15 Watts and each of the two VLSI chips consumes approximately 1.0 Watt. The total power consumption for the GT-EP package is 7.967 Watts with a performance throughput of 10 MFLOPS. The I/O bandwidth of the GT-EP processor is 640 Mbps. The package weighs approximately 320 grams. The characteristics of the GT-EP processor are summarized in Figure 12.

The GT-SP requires 7 custom VLSI chips and 36 off-the-shelf 8kx8 memory chips. The physical and operating characteristics of the chips are shown in Figure 13. Figure 14 shows the packaging layout of the GT-SP processor. The number of MOPS in Figure 13 is based on a frame rate of 10,000 for the GT-VGS gamma suppression chip and 100 for the others. The GT-VSF, GT-VTHR, GT-VCLS, and GT-VCTR chips are capable of operating at 200 frames per second which effectively doubles the MOPS figures for the chips. The total throughput for the Georgia Tech GT-VSP is in excess of 969 MOPS. The only support chips required for the GT-SP processor are memory chips for the GT-VNUC and GT-VTF. The GT-VNUC requires twenty 8kx8 RAM chips and the GT-VTF requires sixteen 8kx8 RAM chips. Each of the RAM chips consumes approximately 0.15 Watts. The total power consumption for the GT-SP processor is 28.6 Watts. At the front-end of the GT-SP processor are sixteen GT-VGS chips which process pixel data at 10,000 frames per second. At this rate each GT-VGS chip provides a bandwidth of 163.8 Mbps with a

Chip	Size (Mil2)	Power (Watts)	#required
GT-VIAG	400x400	1.0	1
GT-VDAG	400x400	1.0	1
GT-VFPU	362x383	2.5	1
GT-VSNI	276x301	0.617	1
Memory (8kx8)	280x280	0.150	19
Total	1,809,600	7.967	23
Package	1"x1"x0.8", 320 grams / GT-EP		
Performance	10 MFLOPS	640 Mbps	

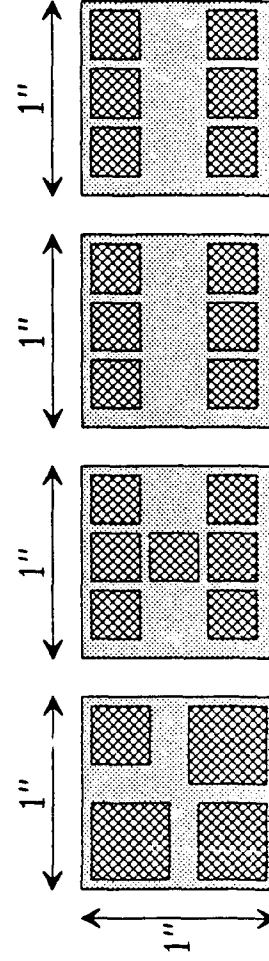


Figure 12. Characteristics of the GT-EP Processor

Chip	MOPS	Size (Mil ²)	Power (Watts)	#required
GT-VGS	20	400x400	1.0	16
GT-VNUC	12	400x400	1.2	1
GT-VTF	23	400x400	1.0	1
GT-VSF	18	378x401	0.8	1
GT-VTHR	53	415x415	1.1	1
GT-VCLS	493	370x370	0.9	1
GT-VCTR	25	395x395	1.1	2
Memory (8kx8)	N/A	280x280	0.4	36
Total	969	4,078,313	28.6	59
Package	1"x1"x2", 800 grams / GT-SP			
Performance	969 MOPS 2,653 Mbps			

..
Figure 13. Characteristics of the GT-SP Processor

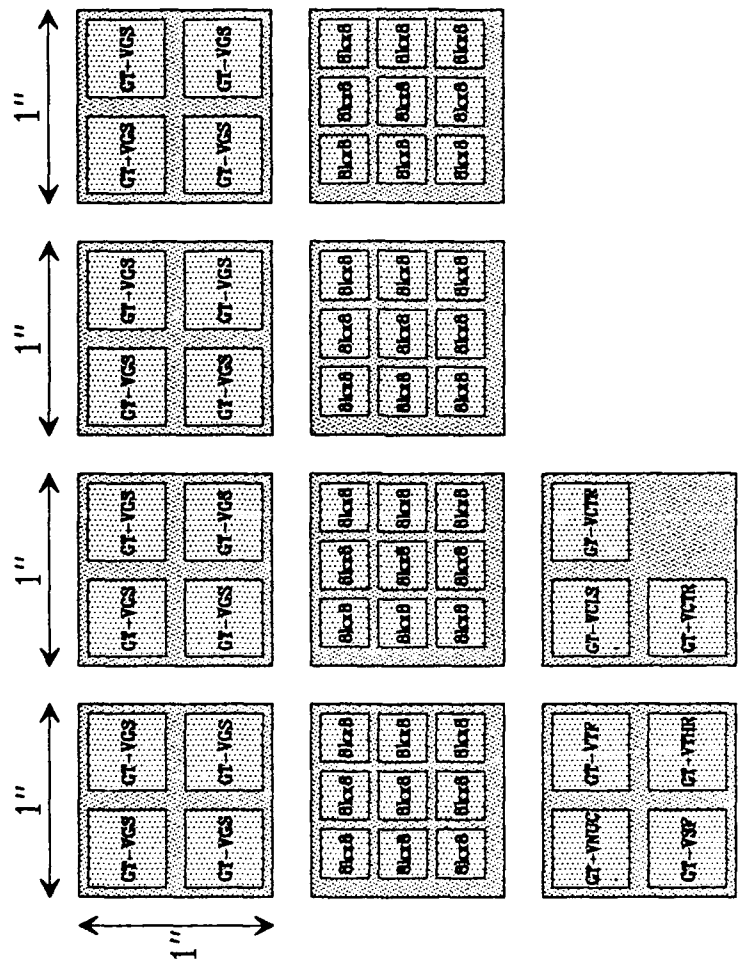


Figure 14. GT-SP Processor Packaging

total 2,621 Mbps for the 16 GT-VGS chips. The back-end of the GT-SP processor has the two GT-VCTR chips which provide cluster information for the object processor. The clustering chips are capable of handling a maximum of $64 \times 64 = 4096$ clusters. Each cluster contains 78 bits of information: area (20 bits), x-coordinate centroid (7 bits), y-coordinate centroid (7 bits), intensity (30 bits), x-intensity centroid (7 bits), and y-intensity centroid (7 bits). At 100 frames per second, the back-end of the GT-SP processor has an I/O bandwidth of $100 \times 64 \times 64 \times 78 = 32$ Mbps.

The GT-VGS chips can be packaged in four 1"x1"x0.2" hybrid boards. The GT-VNUC with its supporting memory chips, requires three boards. The 36 memory chips required for the GT-VNUC and GT-VTF can be packaged in four boards. The GT-VNUC, GT-VTF, GT-VTHR, and GT-VSF can be packaged in one hybrid board. Finally, the GT-VCLS and two GT-VCTR chips can be packaged in another hybrid board. The total number of hybrid boards required for the GT-SP processor is ten, with a combined board space of 1"x1"x2".

The GT-VSM8 is an 8x8 crossbar network with 8 input ports and 8 output ports for data transfer between the various processing units as shown in Figure 1. Each port has a bandwidth of 20 Mbps for a total of 320 Mbps for the GT-VSM8 chip. The GT-VSM8 has a die size of 329x340 mils and consumes 0.8 Watts of power. It can be packaged in a 1"x1"x0.2" board. The characteristics of the network chip are shown in Figure 15.

The parallel processor architecture shown in Figure 1 uses four GT-DP processors, five GT-EP processors (1 as the executive processor, 1 as the I/O processor, and 3 as object processors), and one GT-SP processor. The characteristics of the GN&C processor system are shown in Figure 16. The GN&C system occupies a space of 1"x1"x7" with a power consumption of 91.92 Watts and weighs approximately 2800 grams. The computing power of the GN&C is 76.4 MFLOPS for data and object processing functions. A computing power of 969 MOPS is provided for signal processing on 16 bit fixed points operands. The system I/O bandwidth is 3.6 Gbps (giga bits per second) for data/object processing and 2.6 Gbps for signal processing.

2.6. Technology Progression

The performance, size, and weight calculations are based on VLSI chips designed using 1.25 and 1.0 μ CMOS technology. The GN&C processor can be projected to use 0.5 μ CMOS technology within a five-year time frame which will provide an improvement of a factor of 2 in speed and a factor of 4 in size. A single chip GT-DP processor can be attained by incorporating the GT-VSEQ, GT-VDR, GT-VFPU, and the GT-VSNI chips into a single VLSI chip of approximately 400x400 mil@ size. Four GT-DP processors can then be packaged in a 1"x1"x0.2" hybrid board. The expected power consumption of each GT-DP processor is about 2 Watts with a total of 8 Watts for four GT-DP processors.

Similarly for the GT-EP processor, the GT-VFPU, the GT-VIAG, the GT-VDAG, and the GT-VSNI can be incorporated into a single VLSI chip. Using 8kx32 memory chips with 0.5 μ CMOS technology, the number of memory chips required for the GT-EP processor is five which would enable the packaging of the GT-EP processor in one, 1"x1"x0.2" hybrid board. Five GT-EP processors can be packaged in a

Chip	Size (Mil2)	Power (Watts)
GT-VSM8	329x340	0.8
Package	1"x1"x0.2", 80 grams / GT-DP	
Performance	320 Mbps	

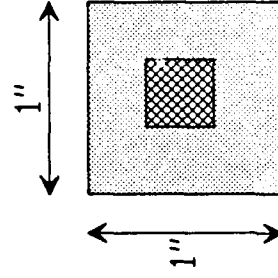


Figure 15. Characteristics of the GT-VSM8 Crossbar Network

Chip	Package	Power (Watts)	#required
GT-DP	1"x1"x0.2"	5.68	4
GT-EP	1"x1"x0.8"	7.96	5
GT-SP	1"x1"x2.0"	28.60	1
GT-VSM8	1"x1"x0.2"	0.80	1
Total	1"x1"x7"	91.92	11
Package	1"x1"x7", 2800 grams		
Performance	76 MFLOPS, 973 MOPS, 6173 Mbps		

Figure 16. Characteristics of the GT-GN&C Processor

1"x1"x1" space. With each 8kx32 chip consuming 0.2 Watts of power, each GT-EP processor would consume about 3 Watts with a total of 15 Watts for five GT-EP processors.

For the GT-SP processor, the chip count for the GT-VGS can be reduced to 4 by incorporating 4 GT-VGS chips into a single VLSI chip. The GT-VNUC, the GT-VTF, the GT-VSF, and the GT-VTHR chips can be incorporated into a single VLSI chip. The GT-VCLS chip and two GT-VCTR chips can be incorporated into a single VLSI chip. In addition, the memory chip count for the GT-VTF, and GT-VNUC can be reduced to nine. Four GT-VGS chips would occupy one 1"x1"x0.2" board. The GT-VNUC-VTF-VSF-VTHR chip, GT-VCLS-VCTR chip, and nine memory chips would occupy two 1"x1"x0.2" boards. The GT-SP processor would then require three boards with a package size of 1"x1"x0.6". Power consumption would be around 2 Watts for each of the VLSI chips and 0.2 Watts for each of the memory chips. Total power consumption for the GT-SP processor would be approximately 13.8 Watts.

With a progression to 0.5 u CMOS technology, the Georgia Tech GN&C processor can be packaged in 1"x1"x2". With a factor of 2 improvement in speed the GN&C processor would then have a performance of 152 MFLOPS for data/object processing and 1946 MOPS for signal processing with an expected I/O bandwidth of 12.3 Gbps. The projected characteristics of the GN&C processor, in 0.5 u technology, are shown in Figure 17.

Chip	Package	Power (Watts)	#required
GT-DP	1"x1"x0.2"	8	1
GT-EP	1"x1"x0.2"	3	5
GT-SP	1"x1"x0.6"	13.8	1
GT-VSM8	1"x1"x0.2"	0.8	1
Total	1"x1"x2"	37.6	8
Package	1"x1"x2", 800 grams		
Performance	152 MFLOPS, 1946 MOPS, 12346 Mbps		

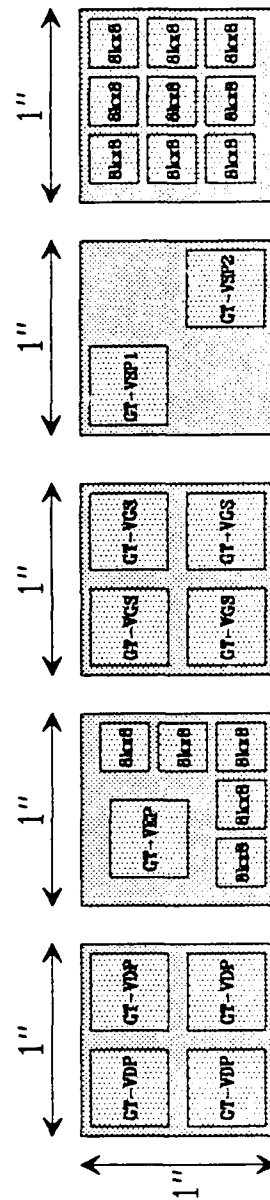


Figure 17. Characteristics of the GT-GN&C Processor with 0.5 u Technology

3. GN&C Processor Software

3.1. Support Software

3.1.1. Compilers

Two Pascal compilers have been written each for the GT-DP and the GT-EP. The two compilers were written so that test programs can be more easily devised in a high level language. They were derived from the Pascal compiler written for the GT-FPP processor [25]. The compilers have been used to compile code for the GT-DP processor module and for the GT-EP multichip simulation module.

Eventually, a C compiler will be written to support the GT-EP and GT-DP. An off-the-shelf validated Ada-to-C compiler had been purchased. The source licence to the back-end of the Ada compiler has also been acquired. This allows the compiler to be tailored to the specific run-time system needs of the GT-EP and GT-DP.

3.1.2. Utilities

A loader for the GT-DP has been written to run on a PC. Loading a program to the GT-DP consists of three phases. In the first phase, a generic program that enables the GT-DP to load constants into the data memory is loaded into the GT-DP instruction memory. In the second phase, data constants data are passed from the host PC to the GT-DP. In the third phase, the program code is loaded into the GT-DP instruction memory. At this point the GT-DP is ready to start program execution.

In order for the host PC to communicate with the GT-DP once the program begins to execute, some basic primitive I/O procedures/functions are needed. These include send and receive procedures/functions for each of the different data types (integer, bitfield, and real) supported by the GT-DP. Other basic routines that are required are "reset processor", "start processor", and "stop processor." A host set of configuration routines were written to allow the loader to configure the various GT-DP VLSI chips.

The loader and the supporting functions will eventually be incorporated into the Integrated Parallel Programming Framework that runs under a Unix operating system.

3.2. Flight Software

The GN&C flight software algorithms will be gleaned from the Exosim 6 DOF missile simulation. The approach is to decide exactly which variables the GN&C processor will have access to during fly-out. The current thinking is that the IMU will provide the input stimuli and the GN&C will calculate thruster direction signals (angle for TVC and bang-bang for FRACS). This approach most likely requires some low fidelity models of the high fidelity Exosim routines.

In the current Exosim code, the auto-pilot algorithm is spread over several subroutines. The Exosim subroutines BAUTO, BGUID and BSTEER contain the TVC auto-pilot and sections of the FRAC auto-pilot. The Exosim subroutine FRACS adds the bang-bang controller for the FRAC thrusters. These routines are executed using integrated states such as velocity and altitude as subroutine arguments. In a real

GN&C, these variables would not be externally available. For a realistic GN&C simulation, the GN&C must calculate all of the required variables internally based only on time and IMU data.

This requirement adds a considerable amount of code to the GN&C auto-pilot. The auto-pilot is responsible for solving AERO, the Exosim aerodynamic force routine, ATMOS, the Exosim atmospheric parameters routine, BTHRST, the Exosim TVC thruster force routine, a small portion of FRCTHR, the Exosim FRAC thruster routine, MASSPR, the Exosim center of gravity and inertia calculations and MISSIL, the Exosim integration routine. This requires, at least to some extent, the entire Exosim 6 DOF to be running in real time on a GN&C processor.

Obviously, code complexity precludes running the Exosim 6 DOF on a current GN&C. Some simplifications must be made to the Exosim 6 DOF. These simplifications must be able to provide the auto-pilot sufficient information at a high enough resolution to guide the missile along the desired trajectory. There are two simplifications that are being investigated.

The first simplification is to replace the table look-up parameters with a simple closed form representation. This closed form representation can consist of a series approximation or a piece-wise linear approximation. The table look-up approach will be used in the GN&C code when a "good" closed form representation can not be found or when the table look-up executes in less time than the closed form representation. This approach can be followed because of the fuzzy nature of many table parameters. Parameters such as atmospheric pressure will vary in the real world so that the controller must be robust enough so that perturbations will not upset the stability of the missile. An inexact parameter estimation can be considered to be a perturbation. Of course the size of the perturbation is of some concern.

The second simplification is to allow the auto-pilot to run on a coarse time scale when compared to the high-fidelity Exosim 6 DOF. The high fidelity Exosim 6 DOF is currently running at a 1ms time step. The GN&C 6 DOF may be able to run at a considerably lower rate and still be able to maintain stability along the desired trajectory. The time issue is tricky for two reasons. The first is that the actuator updates occur on a slower time scale requiring a more robust controller. Secondly, the integrated parameters in GN&C simulation will not be as accurate as the high-fidelity model. This could lead to some divergence from the desired trajectory. The PFP simulations can help to decide the fidelity and time step required for the auto-pilot algorithms in the Exosim 6 DOF. If the processor demands are too high, some other controller algorithms based on artificial intelligence concepts may be required. These types include fuzzy set theoretic and neural network based controllers.

- [1] Ghori, A., *GT-VSEQ: Sequencer Chip, VLSI Design Document*, Special Technical Report CERL008-0040.1, Georgia Tech, July 1988
- [2] Ghori, A., *GT-VSEQ: Sequencer Chip, Design Verification Document*, Special Technical Report CERL008-0041.1, Georgia Tech, July 1988
- [3] Russ, S., *GT-VDR: Instruction and Data Memory Unit, VLSI Design Document*, Special Technical Report CERL008-0050.1, August 17, 1988
- [4] Russ, S., *GT-VDR: Instruction and Data Memory Unit, Design Verification Document*, Special Technical Report CERL008-0051.1, August 17, 1988
- [5] Russ, S.H., *GT-VFPU: Floating/Fix Point ALU Unit, VLSI Design Document*, Special Technical Report CERL008-0030.1, August 17, 1988
- [6] Russ, S.H., *GT-VFPU: Floating/Fix Point ALU Unit, Design Verification Document*, Special Technical Report CERL008-0031.1, August 17, 1988
- [7] Tan, W.S., *GT-VSNI: Serial Network Interface Chip, VLSI Design Document*, Special Technical Report CERL008-0010.1, May 12, 1988
- [8] Tan, W.S., *GT-VSNI: Serial Network Interface Chip, Design Verification Document*, Special Technical Report CERL008-0011.1, May 12, 1988
- [9] Tan, W. S., S. H. Russ, and C. O. Alford, *GT-EP: A High Performance Real-time Processor*, Special Technical Report, CERL008-0170.1, August 1990
- [10] Tan, W.S., *GT-VIAG: Instruction Address Generation, Programming Model*, Special Technical Report, CERL008-0060.1, November, 1988
- [11] Tan, W.S., *GT-VDAG: Data Address Generation, Programming Model*, Special Technical Report, CERL008-0070.1, November, 1988
- [12] Tan, W.S., *GT-VDAG: Data Address Generation, VLSI Design Document*, Special Technical Report, CERL008-0071.1, August, 1990
- [13] Tan, W.S., *GT-VDAG: Data Address Generation, Design Verification Document*, Special Technical Report, CERL008-0072.1, August 1990
- [14] Register, A., *GT-VTF: Temporal Filter, Functional Specification Document*, Special Technical Report, CERL008-0162.1, August 1990
- [15] Ghori, A., *GT-VTF: Spatial Filtering Chip, VLSI Design Document*, Special Technical Report, CERL008-0090.1, March 1990
- [16] Ghori, A., *GT-VTF: Spatial Filtering Chip, Design Verification Document*, Special Technical Report, CERL008-0091.1, March 1990
- [17] Braham, R., *GT-VTHR: Thresholding, VLSI Design Document*, Special Technical Re-

port, CERL 008-0080.1, April 1990

- [18] Braham, R., *GT-VTHR: Thresholding, Design Verification Document*, Special Technical Report, CERL 008-0081.1, April 1990
- [19] Chamdani, J. I., *GT-VCLS: Clustering, VLSI Design Document*, Special Technical Report, CERL008-0100.1, February 1990
- [20] Chamdani, J. I., *GT-VCLS: Clustering, Design Verification Document*, Special Technical Report, CERL008-0101.1, February 1990
- [21] Chamdani, J. I., *GT-VCTR: Centroiding, VLSI Design Document*, Special Technical Report, CERL008-0110.1, February 1990
- [22] Chamdani, J. I., *GT-VCTR: Centroiding, Design Verification Document*, Special Technical Report, CERL008-0111.1, February 1990
- [23] Ghorl, A., *GT-VSM8: 8-point Switching Matrix Chip, VLSI Design Document*, Special Technical Report, CERL008-0020.1, May 1988
- [24] Ghorl, A., *GT-VSM8: 8-point Switching Matrix Chip, Design Verification Document*, Special Technical Report, CERL008-0021.1, May 1988
- [25] Tan, W. S., *GT-FPP: High Speed Floating Point Processor, Compiler Design Document*, Special Technical Report, CERL008-002.1, August 1987