A Hybrid FPGA/DSP/GPP Prototype Architecture for SAR and STAP

Jack M. West, Hongping Li, Sirirut Vanichayobon, Jeffrey T. Muehring, John K. Antonio, and Sudarshan K. Dhall

School of Computer Science
University of Oklahoma
antonio@ou.edu

HPEC 2000
The Fourth Annual Workshop on High-Performance Embedded Computing
September 20-22, 2000
Hybrid FPGA/DSP/GPP Prototype Architecture
Logical Detail

Mercury DSP/GPP Subsystem

Annapolis FPGA Subsystem (F)

Data Source PC

Custom Interface Cables

Annapolis FPGA Subsystem (B)

Data Sink PC
Hybrid FPGA/DSP/GPP Prototype Architecture

Photograph
Communication from Annapolis FPGA (F) to Mercury Interface Design

Annapolis FPGA Subsystem (F)

- **Init**
- **Wait**
- **Write_to_RIN-T**
  - suspend¹
  - buffer_full²
  - buffer_empty³
- **Read_from_HOST**

- **suspend¹**
- **buffer_full²**
- **buffer_empty³**

Mercury Subsystem

- **Init RIN-T**
- **Wait_for_data**
  - not_empty
  - Strobe
  - Valid
  - Suspend
- **complete**
- **Determine_Dest_CN**
- **Send_Data**
  - Create_DX_transfer

¹ Suspend from the RIN-T
² FPGA memory buffer is full
³ FPGA memory buffer is empty

*Peak throughput achieved to date: (15 MHz) \times (4 \text{ Bytes}) = 60 \text{ Mbytes/sec}*
Communication from Mercury to Annapolis FPGA (B)

Interface Design

**Mercury Subsystem**

1. **Init ROUT-T**
2. **Wait_for_data**
3. **Pack_Data**
4. **Send_Data_to_ROUT-T**
5. **Replicate_Data**
6. **Create_DX_transfer**

**Annapolis FPGA Subsystem (B)**

1. **Init ROUT-T**
2. **Read_from_ROUT-T**
3. **Write_to_Host**
4. **Wait**
5. **valid**

- **32 Data**
- **valid**
- **valid**
- **valid**
- **valid**
- **Strobe**
- **Valid**
- **Suspend**
- **buffer_empty**
- **buffer_full**

*Peak effective throughput: \((33 \text{ MHz}) \times (4 \text{ Bytes}) \times (1/4) \times (30/32) = 31 \text{ Mbytes/sec}\)*

replication factor  
packing factor
Data Packing/Unpacking Overview
For Mercury-to-Annapolis Communication
# Data Packing/Unpacking Algorithm Detail

For Mercury-to-Annapolis Communication

<table>
<thead>
<tr>
<th>Pack Data</th>
<th>Unpack Data</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="8-bit original data" /></td>
<td><img src="image" alt="8-bit filtered data with 2-bit encoded counter" /></td>
</tr>
<tr>
<td><img src="image" alt="8-bit packed data w/ 2-bit encoded counter" /></td>
<td><img src="image" alt="8-bit original data" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Replicate Data</th>
<th>Filter Data</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="8-bit packed data" /></td>
<td><img src="image" alt="8-bit filtered data with replication" /></td>
</tr>
<tr>
<td><img src="image" alt="8-bit packed data w/ replication" /></td>
<td><img src="image" alt="8-bit filtered data with 2-bit encoded counter" /></td>
</tr>
</tbody>
</table>

1 The true data size is 32-bit. This example uses 8-bit data for illustration purposes only.
Streaming Parallel RT_STAP on Mercury Subsystem

Input Manager (SHARC)  
Output Manager (SHARC)

Processing CNs (PowerPCs)

RINT  
2 – 4Kx18 FIFOs

Distribute Input Data Cube

CN1  
SMB (data)

CN2  
SMB (data)

CN7  
SMB (data)

CN8  
SMB (data)

Gather Output Data Matrix

ROUT  
2 – 4Kx18 FIFOs

SMB (data)  
sync
Parallel RT_STAP on Mercury Subsystem*

Pulse Compress (range dimension whole)

Doppler Filter (pulse dimension whole)

QR Decomposition (channel-range seq. planes whole)

CN1
CN2
CN7
CN8

Input Data Cube
Re-Partition Data Cube
Re-Partition Data Cube
Output Data Matrix

Space-Time Diagram for Parallel RT_STAP
Using 8 PPC CNs for Processing and 2 SHARC CNs for I/O

Input Data Cube 1

Input Data Cube 2

Output Data Matrix

SHARC (Input)

CN1

CN2

CN3

CN4

CN5

CN6

CN7

CN8

SHARC (Output)

comm. time

idle time

t=0

t=4s

T=4.5s
Throughput Requirements for Medium Case Parallel RT_STAP
Using 8 PPC CNs for Processing and 2 SHARC CNs for I/O

<table>
<thead>
<tr>
<th>Function</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribute Input Data</td>
<td>4 sec</td>
</tr>
<tr>
<td>Pulse Compress</td>
<td>299.48 msec</td>
</tr>
<tr>
<td>First Rotation</td>
<td>21.18 msec</td>
</tr>
<tr>
<td>Doppler Filter</td>
<td>25.32 msec</td>
</tr>
<tr>
<td>Second Rotation</td>
<td>112.48 msec</td>
</tr>
<tr>
<td>QR Decomposition</td>
<td>99.36 msec</td>
</tr>
<tr>
<td>Gather Output Data</td>
<td>23 msec</td>
</tr>
<tr>
<td><strong>Total Time</strong></td>
<td><strong>4.5 sec</strong></td>
</tr>
</tbody>
</table>

Input Data Size = 16 × 64 × 1920 × 2 = 4 MBytes
Output Data Size = 64 × 480 × 8 = 0.25 MBytes

Input Throughput = 4 Mbytes/4.5 sec
= 0.89 Mbytes/sec

Output Throughput = 0.25 Mbytes/4.5 sec
= 0.056 Mbytes/sec
SAR Processing Flow*

Input Data

Fix-to-Float

Digital I/Q (real-to-complex)

Pulse return
N range cells

Pulse Compression

Range-Compressed
Pulse return
N range cells

Corner-Turning
Double-Buffer

Azm. Compression
-Fast Convolution (sectioned)

Magnitude

Output Image Buffer

N=2048

K: Pulse Number =512

Data Distribution for Parallel SAR Processing on Mercury
Using 6 PPC CNs for Processing and 2 SHARC CNs for I/O

Input (Odd Pulses From SHARC CN 1)

CN 2 Input Buffer
1, 2, ……, …… 2048
(2048 range gates)

CN2 Range Processing

CN 2 Output Buffer
1 512 1024 1536 2048

CN2 DMA CN3 DMA
CN2 DMA CN3 DMA
CN2 DMA CN3 DMA
CN2 DMA CN3 DMA
CN2 DMA CN3 DMA

CN4 Input Buffer
CN5 Input Buffer
CN6 Input Buffer
CN7 Input Buffer

CN 4 Corner Turn
CN 5 Corner Turn
CN 6 Corner Turn
CN 7 Corner Turn

CN 4 Double-Buffered Memory
(512 * 1024 double complex data)

CN4 DMA
CN 5 DMA
CN 6 DMA
CN 7 DMA

CN 4 Azimuth Processing
CN 4 Output Buffer

CN 4 DMA
CN 5 DMA
CN 6 DMA
CN 7 DMA

4 * 512
* 512

CN 8 (SHARC) Output Image Buffer
Space-Time Diagram for Streaming Parallel SAR Processing
Using 6 PPC CNs for Processing and 2 SHARC CNs for I/O

CN1
(input)

CN2

CN3

CN4

CN5

CN6

CN7

CN8
(output)

odd pulses

even pulses

comm. time

idle time

(512 pulses)

(512 range gates)

(512 range gates)

(512 range gates)

(512 range gates)

(512 range gates)

(512 range gates)

(2048 range gates)

(2048 range gates)

(256 pulses)

(256 Pulses)

(256 Pulses)

(256 Pulses)

(256 Pulses)

(256 Pulses)

(256 Pulses)

(256 Pulses)

0x0

0x0

0x0

0x0

0x0

0x0

0x0

0x0

et=0

et=5.6s

et=11.2s

et=16.8s
Streaming Parallel SAR Processing Throughput Requirements
Using 6 PPC CNs for Processing and 2 SHARC CNs for I/O

<table>
<thead>
<tr>
<th></th>
<th>CN1</th>
<th>CN2</th>
<th>CN4</th>
<th>CN8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Data Size</td>
<td>512 × 2</td>
<td>2 × 2032</td>
<td>2 × 2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>= 4 MBytes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Input Throughput</td>
<td>4 MBytes/5.6 sec</td>
<td>= 0.71 MBytes/sec</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Output Data Size</td>
<td>512 × 2048</td>
<td>× 2 × 4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>= 8 Mbytes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Out Throughput</td>
<td>8 MBytes/5.6 sec</td>
<td>= 1.42 MBytes/sec</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>