

# A Reconfigurable Computing Platform for Detection and Direction Finding of Frequency Hopping Signals

K. Haji, and B. Brown

Southwest Research Institute, P.O. Drawer 28510, San Antonio, TX 78228-0510

## ABSTRACT

SwRI is developing a reconfigurable computing platform (RCP) for detection and direction finding (DF) of frequency hopping signals using COTS FPGA-based boards derived from NSA's SPLASH2 technology. Concurrent DF Processors process different sub-bands of the tuned spectrum as tasked by the detection alarms. The architecture supports wideband detection and DF processor functions. The Detection Processor is comprised of eight FPGAs, each supporting one Detection Macro, while the DF Processor is comprised of eight FPGAs, only six of which implement the DF Macros. The number of concurrent Detection and DF Macros depends on the DF algorithm used, the optimization methodology (area vs. speed), and the capacity of the FPGA. The RCP is estimated to provide at least an order of magnitude performance improvement over DSP-based solutions.

## I. INTRODUCTION

This paper considers implementation of detection and direction finding (DF) algorithms on a reconfigurable computing platform (RCP) to achieve real-time acquisition and tracking of frequency hopping targets. The RCP is at the heart of a wide band tactical surveillance system named AFH-TAS (Advanced Frequency Hopper Target Acquisition System) which is being developed at Southwest Research Institute under an internal R&D project.

The AFH-TAS, as shown in Figure 1, performs three tasks in real time: *detect*, *direction find*, and *track* frequency hopping targets. The detection and DF tasks are computationally very expensive, and usually require multiple COTS boards, each with quad or octal DSPs on it. By mapping the detection and DF algorithms to the RCP, at least an order of magnitude performance improvement over traditional COTS DSP-based solutions is expected to be achieved. This has significant ramifications on the cost and SWAP (size, weight, and power) of the wideband system. SWAP is especially important in airborne applications affecting mission planning. The AFH-TAS is expected to be a single VME-chassis solution due to SWAP reduction achieved through RCP. The AFH-TAS specifications are shown in Table 1.

### A. The RCP Advantage

The RCP provides the speed of application specific hardware with the flexibility of software. A library of

algorithms suitable for different tactical environments can be developed that are mapped to the RCP's architecture. The algorithms are converted to VHDL code that eventually gets translated to a gate-level netlist. The netlist configures the FPGAs populating the RCP. The configuration files are downloaded via network to the system controller on the chassis. Algorithm modifications can be done online and downloaded to the RCP.

Another advantage is scalability. The same design can target faster hoppers by swapping the current RCP with its later versions that incorporate higher density FPGAs, without changing the system architecture, and with minimum system software impact.

Table 1: AFH-TAS Specifications

Spectrum Coverage	VHF/UHF
Target	Common FH signals
Environment	Multiple hopper targets
Standard Mission	Energy detection, direction finding, and tracking
Optional Mission	Dehop and copy
Instantaneous BW	4-20 MHz
Frequency Coverage	Wide (proprietary information)
Hops/Sec	Up to several thousand per second (proprietary information)
Design	All COTS
RTOS	Tornado/VxWorks

## II. THE RCP

The RCP functions are depicted in Figure 2. The spectrum data streams from the FFT engines are delayed long enough for the Detection Processor to generate the alarm data. The DF Processor receives the alarms and the delayed FFT data streams and extracts bearing information from them. It then sends bearing and alarm information to the hopper tracker engine running on the host.

The RCP is comprised of two WildChild™ boards by Annapolis Micro Systems (AMS), Annapolis, Maryland. One WildChild™ is used as the Detection Processor, and the other as the DF Processor. The WildChild™ is an implementation of the SPLASH2 technology developed by NSA and licensed to AMS.

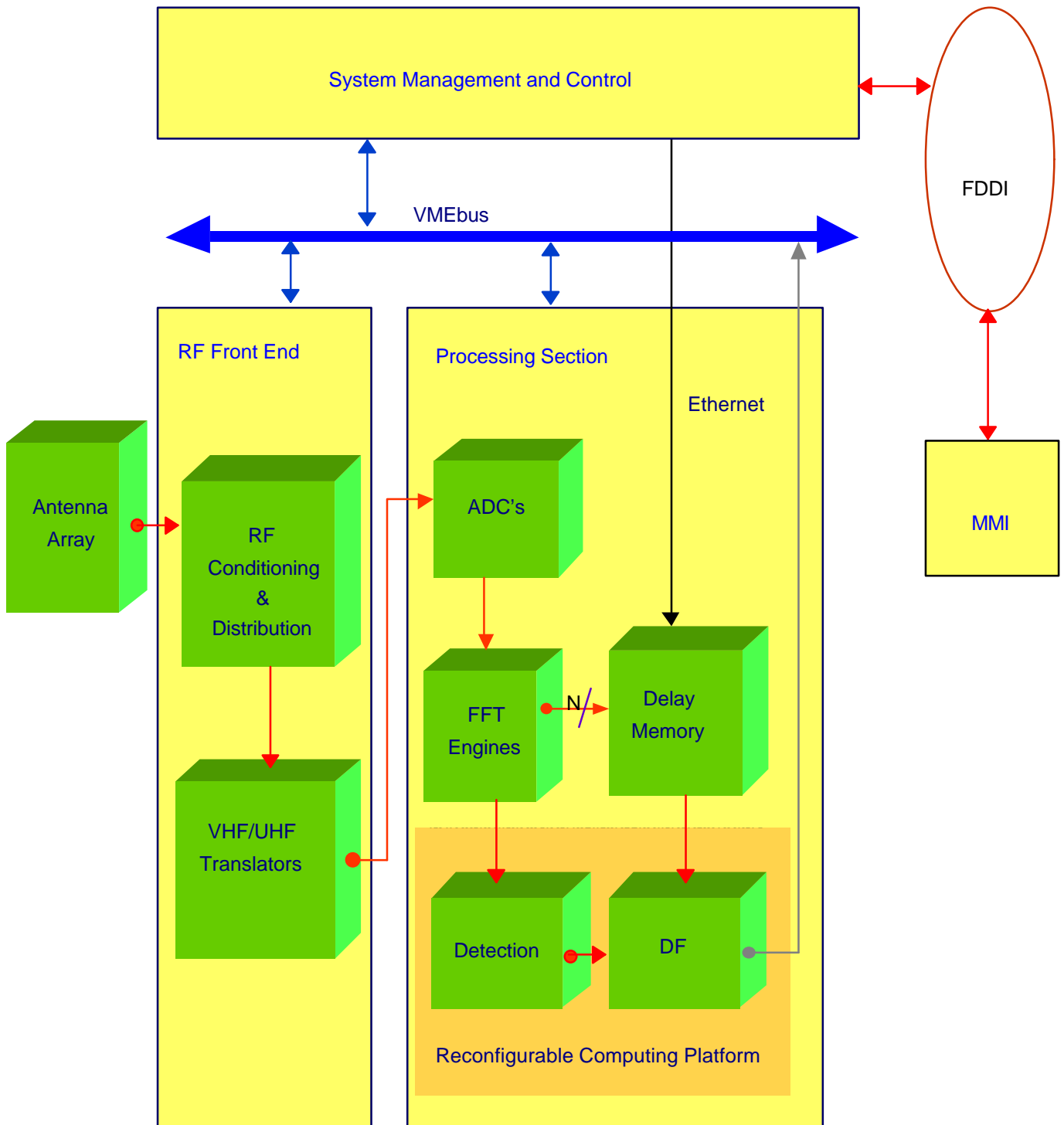


Figure 1: AFH-TAS Block Diagram

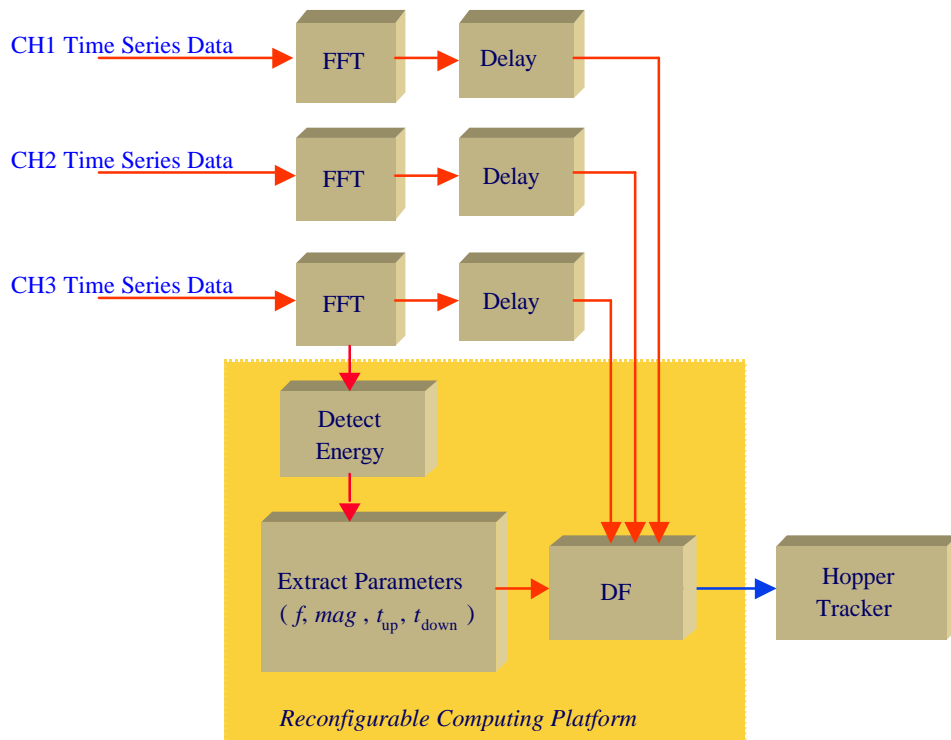


Figure 2: The RCP Function

### A. WildChild™ Features

The Wildchild™ board and its architecture are depicted in Figures 3 and 4. As can be seen, the Wildchild™ is made up of nine processing elements (PE<sub>0</sub>-PE<sub>8</sub>) attached to a crossbar switch via ten 36-bit bi-directional ports (two for PE<sub>0</sub> and one for each of PE<sub>1</sub> to PE<sub>8</sub>).



Figure 3: The Wildchild™

Each PE<sub>n</sub> is connected to PE<sub>n-1</sub> and PE<sub>n+1</sub> (n = 2, ..., 7) via a 36-bit bi-directional systolic bus. PE<sub>1</sub> and PE<sub>8</sub> are each terminated to a systolic connector for multiple WildChild™ configurations. PE<sub>0</sub> is terminated to a SIMD (single instruction, multiple data) connector.

Each PE, as shown in Figure 5, has a Xilinx XC4000 family FPGA (XC4028EX-3 in our application), 512K Bytes (256K x 16) Shared Port RAM (SPR) except for PE<sub>0</sub> that has 1 Mbytes (512K x 16) SPR, and a SPR Memory Controller. Three FIFO's facilitate communication between VMEbus on one side and PE<sub>0</sub>, PE<sub>1</sub>, and PE<sub>8</sub> on the other.

In the AFH-TAS RCP, the two systolic and one SIMD connectors are used to bring in the frequency spectrum data to the DF Wildchild™. On the Detection Wildchild™ only the SIMD connector is used.

### B. WildChild™ Development Environment

For AFH-TAS, the Windows™ NT is used as the OS environment to develop VHDL code. AMS provides PE<sub>0</sub> and PE<sub>n</sub> core design templates. The user-developed VHDL modules implementing the algorithms are instantiated within the AMS templates. Other tools used are:

- For simulation, ModelSim™ VHDL Simulator by Model Tech Inc., Beaverton, OR;
- for Synthesis, Synplify® by Synplicity, San Jose, CA;
- for RTL place and route verification, M1-Foundation Series by Xilinx, San Jose, CA.

The RCP design cycle is shown in Figure 6.

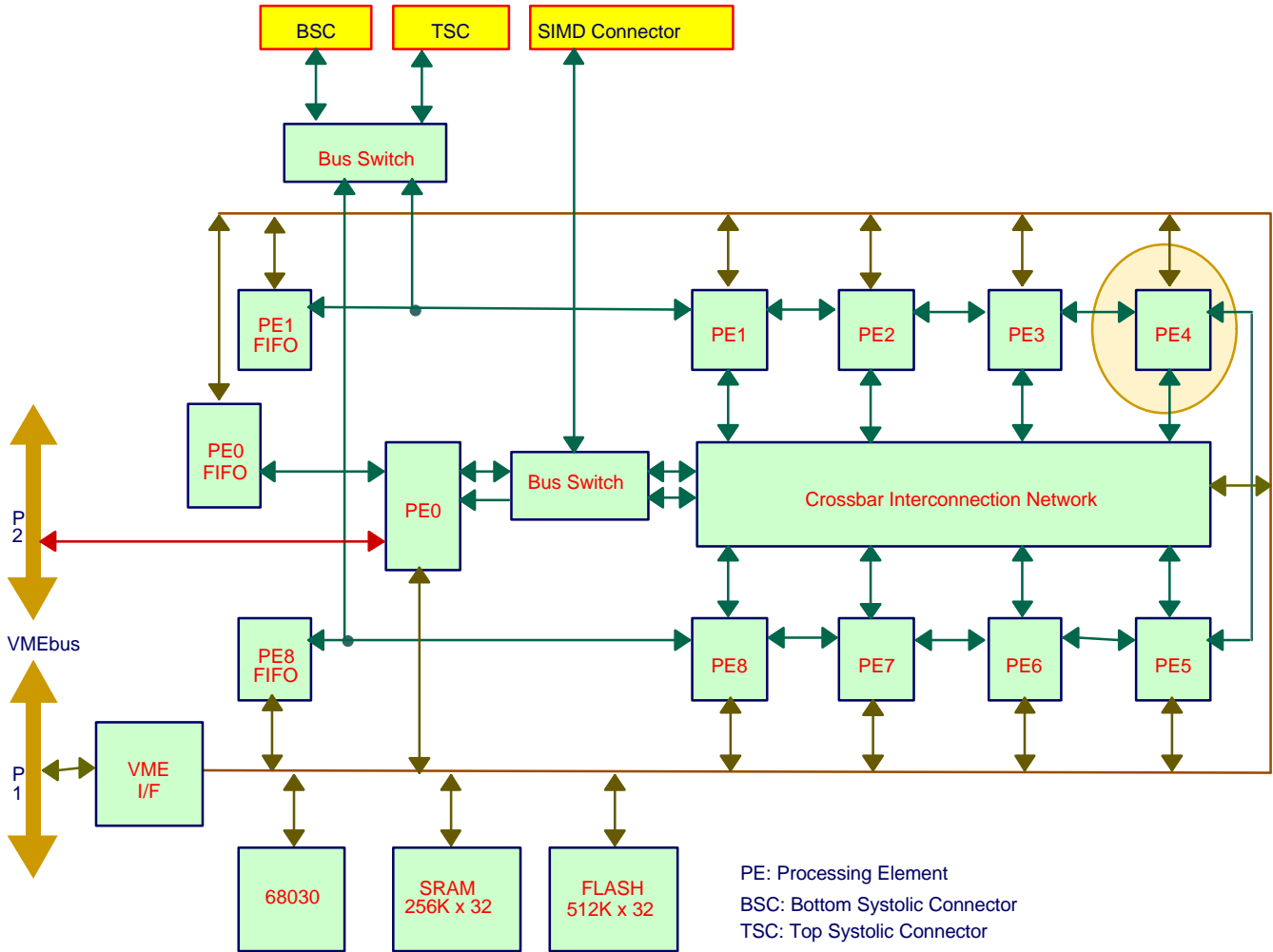


Figure 4: The Wildchild™ Architecture

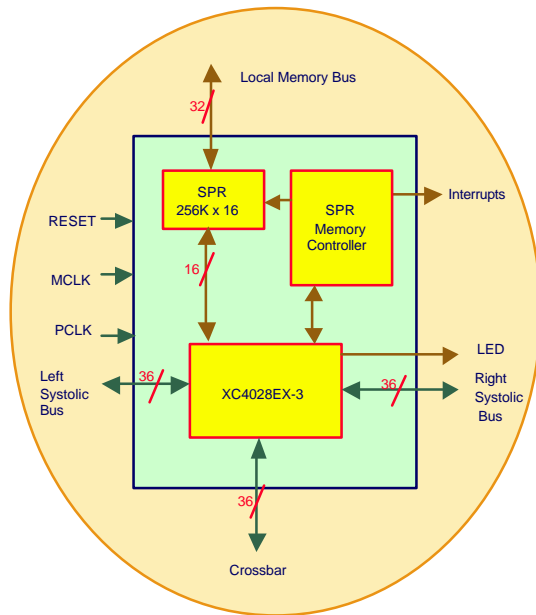


Figure 5: PE Architecture

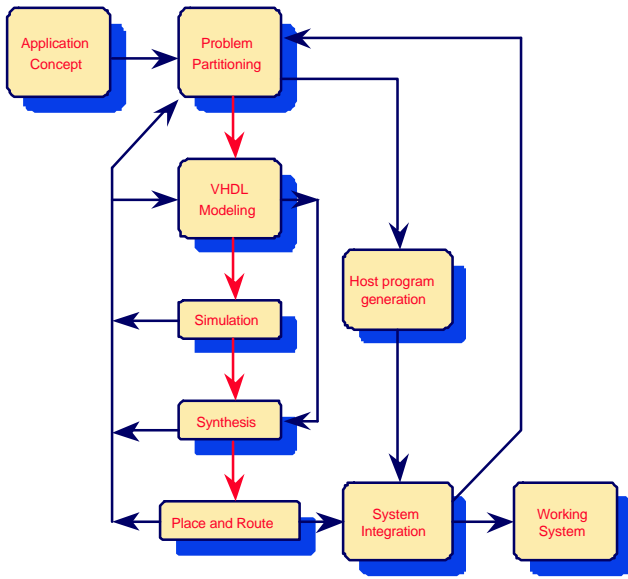


Figure 6: Design Flow Chart[1]

### III. RCP DESIGN METHODOLOGY

Since the application requires a data flow design, pipelines are implemented to achieve one math function (multiply, add, divide) per clock cycle after a few clock cycles of latency up-front. The algorithm is partitioned in a parallel processing approach to create identical designs, with an eye towards the gate count constraint of each PE. Multiple  $PE_n$  have the same configuration, which yields a reduction in the time spent on synthesis, placement, and routing.

#### A. Detection

The Detection Flow Chart is shown in Figure 7. During each spectrum revisit, short and long term magnitude averages are calculated on a frequency slice basis. Signal energy events (time-up and time-down) are determined, and a single bin alarm is formed. Alarms are grouped together and reported to the DF Processor.

To implement the detection algorithm, the starting bandwidth is split into sub-bands. Identical Detection Macros are created to process the sub-bands in parallel. The Detection Macros are optimized to fit as many of them as possible in each  $PE_n$ . The width of the sub-band is the starting bandwidth divided by the total number of Detection Macros.

Each Detection Macro “sees” the same FFT stream, responds only to the sub-band allocated to it, detects energy events, creates alarms, and delivers alarms to  $PE_0$ .  $PE_0$  subsequently sends the alarms to the DF Processor.

The Detection Processor is shown in Figure 8.

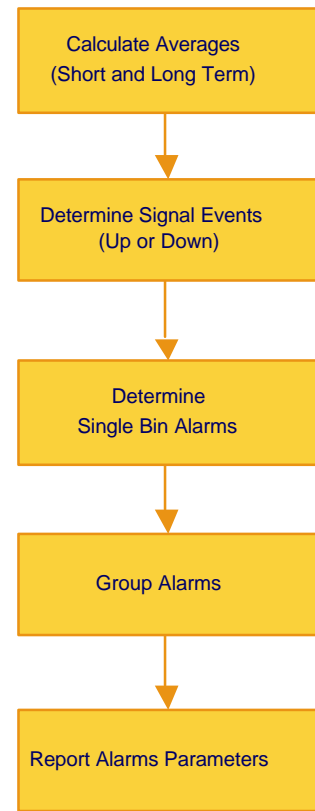


Figure 7: Detection Flow Chart

#### B. DF

Implementation of the DF algorithm follows the same methodology as that of detection. The design is partitioned to create multiple identical macros. Processing common to all DF Macros are tasked to  $PE_1$  and  $PE_8$ . DF Macros are area optimized to fit as many of them as possible in each  $PE_n$  ( $n = 2, \dots, 7$ ).

Each DF Macro “sees” the delayed version of the same FFT stream the Detection Macros see. An Alarms Arbiter tasks each DF Macro with a different alarm from the Detection Processor. The DF Macro then calculates angle of arrival (AOA) for the signal in the FFT stream with the matching alarm, and reports it to the host. The DF Processor is shown in Figure 9.

#### C. CLB Requirement

Mapping of the detection and DF algorithm to the current architecture of Wildchild™ posed some interesting challenges. The board design is optimized for image processing applications, with systolic connectors acting as gateways to other Wildchild™ boards. To achieve our goal, we had to follow a data flow approach, which meant we exhausted all the data connectors of the Wildchild™ for bringing in the frequency data streams. Thus, we were restricted to one board for detection and one for DF. Furthermore, the fact that the memory in each PE was a

shared port RAM and not a true dual ported RAM impacted

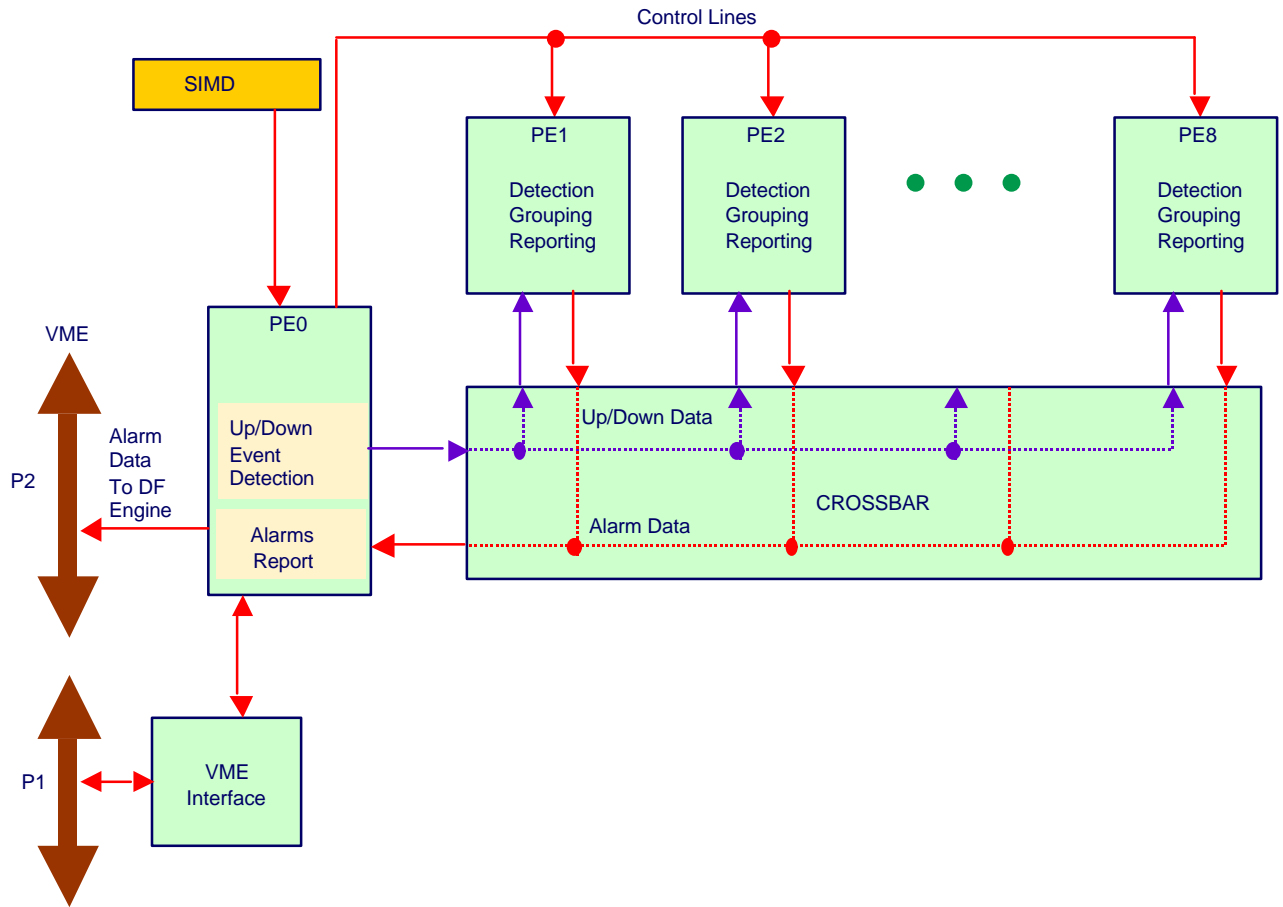


Figure 8: The Detection Processor

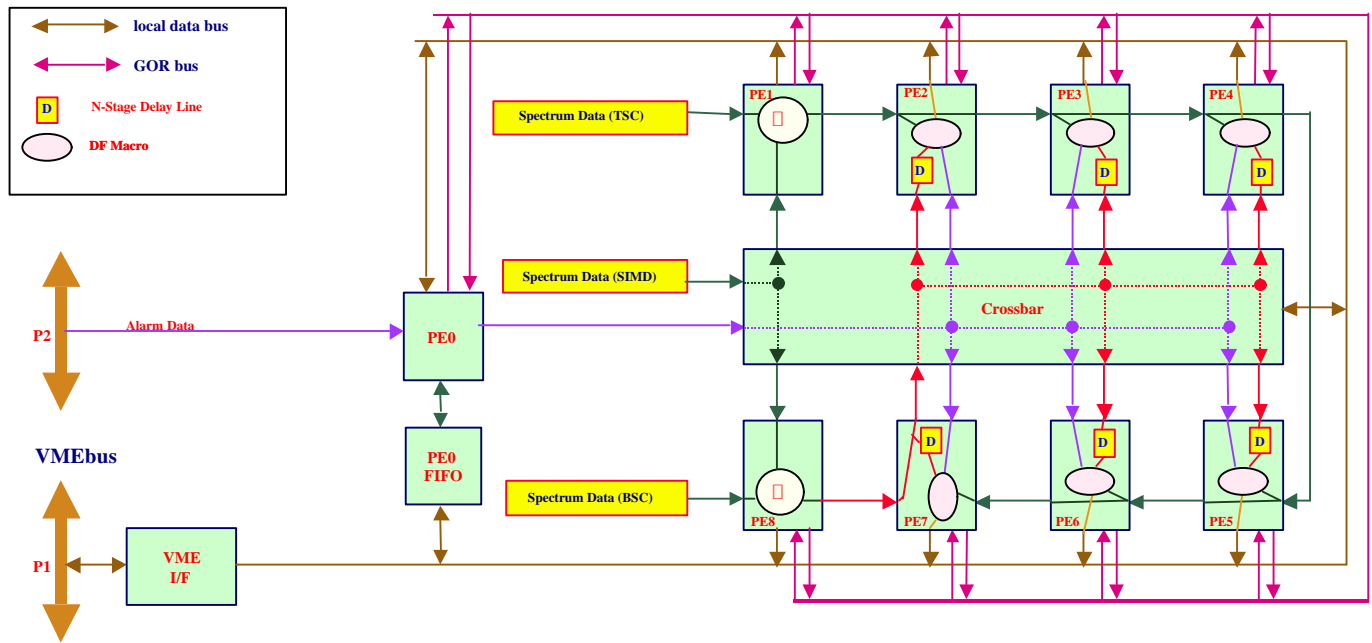


Figure 9: The DF Processor

our design. Finally, the gate count constraint in each PE limited our dynamic range and maximum throughput. We are hopeful the future iterations of the Wildchild™ design address these concerns. We are already aware of the future availability of AMS Wildchild™ boards with XC4085 FPGAs on them which effectively triples our gate resources.

Table 2 shows the CLB requirements for typical math and DSP functions. Since one detection or DF Macro uses about 700 or 900 CLBs respectively, first generation AFH-TAS RCP using XC4028 (1024 CLB) can implement one DF or Detection Macro per PE. With the next generation system using XC4085 (3136 CLB), we should be able to fit four DF or Detection Macros per PE.

#### D. Performance Comparison

A typical FH detection processor may look at 20,000 2K complex FFTs per second, that is 20.4 Million FFT bins per second (MBPS). A 'C40 processor running at 50 MHz can only process 200,000 MBPS. So, it would take over 100 'C40's running in parallel to handle the computation. Ignoring the enormous task of data distribution and using octal C40 COTS board still results in 13 boards. The SwRI AFH-TAS RCP uses one Wildchild™ running at 25 MHz to do the same job. Even with the latest TI 'C6701 DSP boasting a GFLOP at 200 MHz, the task requires more than one board. Current COTS C6701 boards have at most four DSPs running at 167 MHz. This translates into two boards to match the performance of one Wildchild™. Again, considering data distribution issues, a COTS quad C6701 board may not achieve the full potential of its DSP performance. See Table 3.

Table 2. CLB Requirements[2]

Function	Number of CLBs
16 x 16 Parallel Multiplier	213
16 x 16 Constant Coefficient Multiplier	83
32-bit Adder	17
32-bit Accumulator	17
2-input 16-bit Multiplexer	8
16 Word (16 x16) LUT	16
256 Byte LUT	64
256 x 16 FIFO	286
N-bit Counter	(N/2) + 1
N-bit Register	N/2

N-bit Comparator	N
16-bit 1 to 4 Stage Delay Element	9
16-bit 5 to 7 Stage Delay Element	11

Table 3: Hardware Performance Comparison for Detection

25MHz Wildchild™	Octal 50 MHz C40	Quad 167 MHz C6701
20.4 MBPS	1.6 MBPS	13.4 MBPS
1 Board	13 Boards	2 Boards

#### IV. SUMMARY

Use of an RCP in real time signal processing applications can produce a design that is at least an order of magnitude faster than many COTS DSP-based boards. SwRI's internally funded AFH-TAS is exploiting an RCP to create a real time wide band tactical surveillance system against frequency hoppers. The system is scalable both in functionality and performance as faster and denser FPGAs are supported on the AMS RCP.

#### ACKNOWLEDGEMENTS

The authors wish to acknowledge contributions of Mike Brysch and Jennifer Alvarez to the Detection Processor. Also we are grateful for the generous support we have received and continue to receive from Jane and Bob Donaldson and their staff at Annapolis Micro Systems.

#### REFERENCES

- [1] AMS WILDFIRE™ Training Manual, AMS Inc., Annapolis, Maryland, 1998
- [2] Xilinx Core Solutions Data Book, Xilinx, Inc., San Jose, CA, 1998

#### ACRONYMS

<b>AMS:</b>	Annapolis Micro Systems
<b>AOA:</b>	Angle of Arrival
<b>CLB:</b>	Configurable Logic Block
<b>COTS:</b>	Commercial Off The Shelf
<b>DF:</b>	Direction Finding, Direction Finder
<b>DSP:</b>	Digital Signal Processing
<b>FFT:</b>	Fast Fourier Transform
<b>FDDI:</b>	Fibre Digital Data Interface
<b>FPGA:</b>	Field Programmable Gate Array
<b>MBPS:</b>	Million Bins Per Second
<b>MMI:</b>	Man Machine Interface

**NSA:** National Security Agency  
**PE:** Processing Element  
**RCP:** Reconfigurable Computing Platform  
**RTL:** Register Transfer Level  
**SIMD:** Single Instruction Multiple Data