Hardware solutions for radar processing
FPGA / GPU / CPU Trade-offs

Sondra Workshop in Singapore
March 21th 2012
Hardware components for radar processing

FPGA

GPU

CPU
## Characteristics of Candidate Embedded Products

<table>
<thead>
<tr>
<th></th>
<th>AMD Phen. II 1090T CPU</th>
<th>AMD Radeon E6760 GPU</th>
<th>NVIDIA GeForce GT240 GPU (GRA 111)</th>
<th>Xilinx Virtex-6 SW475T FPGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP GFLOPs</td>
<td>153.6</td>
<td>576</td>
<td>250</td>
<td>550</td>
</tr>
<tr>
<td>GFLOPS / W</td>
<td>1.23</td>
<td>16.5</td>
<td>7.1</td>
<td>13.7</td>
</tr>
<tr>
<td>GFLOPS / $</td>
<td>0.54</td>
<td>N/A</td>
<td>2.7</td>
<td>0.14</td>
</tr>
</tbody>
</table>
Nowadays ADCs can sample signals at rates up to a few GigaSamples per second

- signal sampling on carrier frequency or at the first intermediate frequency in the radar receiver

Often two configurations are encountered on modern versatile radars (different modes at different times)

1. Wideband + a few channels (1 to 4)
2. Narrowband + High dynamic + numerous channels (10 to 100 or even 1000 for the full digital radar)

Need to reduce the data rate
- Real time digital beamforming
- Real time filtering and decimation
A digital receiver performs the same function as its analog counterpart, the (super)heterodyne receiver.

- Consists in digitizing a carrier or an Intermediate Frequency (IF), by mean of high speed ADC, and then to digitally down convert it (FIR filtering + decimation).
- Data at the output of the digital receiver are in baseband, with an I&Q encoding, at much lower datarate.
Benefits of Digital Receiver

Benefits
- Cost effective compared to analog solution
- I&Q are well balanced, as they are digitally filtered
- Highly reconfigurable

BUT
- Requires highly skilled people to program FPGAs
Why GPU for radar processing instead of CPU?

**Potential benefits**
- Best ratios Gflops/W, Glops/$
- 500 Gflops for 200 W on Tesla C2075
- Cost (~ 2 K$)
- No ITAR issue (mass market)
- 100% costs and open source
- Scientific library available (CuBLAS for linear algebra)
- Existing programming environment (CUDA)

**Open questions**
- How to formalize radar processing for GPU efficient implementation?
- What effective computing power for which radar processing?
CPU / GPU Comparison

- Fast on iterative calculations
- Low level parallelization tasks

- High performance on massively parallel algorithms
- High level parallelization tasks

**CPU**

- e.g. CPU: Intel X5670 (6 cores)
  - 70 Gflops (double precision)
  - 0.7 GFlops per Watt

**GPU**

- e.g. GPU: NVIDIA Tesla C2075
  - 500 Gflops (double precision)
  - 2.5 GFlops per Watt
Current supercomputer (100+ PowerPC G4 processors) is being replaced by a single GPU card.

Mercury SC
TDP > 8 kW

Tesla C2075
Price = 2.5 K$ (2011)
TDP = 215 W
Application case 1 at Onera: space surveillance ground radar

Pulse compression + Doppler Processing

Time (ms) - Lower is better

- 100 PPC G4
- AMD Opteron 6140*
- Nvidia C2050

16384 Doppler filters
100 receivers
20 Doppler rate
→ 1.3Gflops

Performance Speedup GPU/CPU = x7.2

* All 8 cores active

* All 8 cores active
Application case 1 at Onera: space surveillance ground radar

Digital beamforming + detection

Performance Speedup GPU/CPU = x16.5

* All 8 cores active
Rugged GPGPU product available for defense and aerospace market

Increasing functional capability

Decreasing size, weight & power (SWaP)

CPU only = 16 GFlops peak, 60 W, 0.27 GFlops/W
CPU+GPU = 250 GFlops, 100 W, 2.5 GFlops/W

MAGIC 1

GE-IP’s rugged GRA111
NVIDIA GeForce GT240 GPU

Courtesy of GE Intelligent Plateforms
Application case 2 at ONERA: STAP on Airborne Radar

**Reference Processing**

**Outrescent Pattern Solution**

$$w^H = (w_q^H \Gamma_{\text{th}} w_q) \cdot \frac{w_q^H \Gamma_{\text{th}} \Gamma^{-1} \Gamma_{\text{th}} w_q}{w_q^H \Gamma_{\text{th}} \Gamma^{-1} \Gamma_{\text{th}} w_q}$$

$$w_q^H = [11...1 000...000]$$

N times \((N-1)M\) times
Double precision matrix inversion

Performance Speedup GPU/CPU = \times 6.0

* All 4 cores active

** DP Matrix inversion
Application case 3 at Onera: SAR Processing

Processing time for one SAR image

<table>
<thead>
<tr>
<th>System</th>
<th>Time (mn)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2*Intel X5650</td>
<td>25</td>
</tr>
<tr>
<td>Nvidia C2050 (+1*Intel X5650)</td>
<td>5</td>
</tr>
</tbody>
</table>

To reach Tesla C2050’s performances, **11 CPUs** are needed!
Comparaison GPU/CPU at same performances:

Reaching the same performances with CPU will cost and consume 5 time more!
**Summary**

- **FPGA** are the only solution to face high data rate
  - Broadband, data rate reduction (DDC), performances
  - Highly complex programming, hard to modify
  - Cost

- **CPU** does efficiently sequential work (and controls the GPU)
  - Easy programming, portability, easy algorithms evolution
  - Fast on iterative / non easy parallelizable operations (CFAR, …)
  - Performances, electric consumption, cost

- **GPU** performs very well on huge parallel tasks
  - Performances on parallel processing, easy algorithms evolution
  - Efficiency, small size
  - Data transfers, iterative operations