RTI in Hardware

Robert Martí
Contents

- RT Image Processing
- Platforms
- FPGAs
The time between the presentation of a set of inputs and the appearance of all the associated outputs.
Real-Time Image Processing

Digital Signal in → Real-Time Digital Processing → Digital Signal out

Example:
- Processor clocked at 120 MHz and can perform 120MIPS
- 120MHz / 3 Mpixels = 40 Instructions per pixel

<table>
<thead>
<tr>
<th>Format</th>
<th>Lines/Frame × Pixels/Line × Frames/Second</th>
<th>Sampling Rate (million pixels per second)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIF (videoconferencing)</td>
<td>$360 \times 288 \times 30 = \text{3}$</td>
<td>$3$</td>
</tr>
<tr>
<td>CCIR (TV)</td>
<td>$720 \times 576 \times 30 = \text{12}$</td>
<td>$12$</td>
</tr>
<tr>
<td>HDTV</td>
<td>$1280 \times 720 \times 60 = \text{60}$</td>
<td>$60$</td>
</tr>
</tbody>
</table>
What can be done to achieve RTI?

- Faster hardware: 120 MHz to 2.53 GHz
  
  $2.53 \text{GHz} / 3 \text{ Mpixels} = 840 \text{ Instructions per pixel}$

- Hardware architectures
  - Sequential vs Parallel.
    - What would be the $\text{I}pp$ if we could do 2 instructions in parallel? And $n$?
  - SIMD – MIMD – MISD
  - GPPs – DSPs - FPGAs
RTI data characteristics

- Large number of samples being continuously fed to the system (samples or blocks).
- Repetitive Operations:
  - The same operation being applied to different set of samples
  - Parallel processing
- Vector and Matrix Operations
- Real time operations
Example: Digital Filtering

- The basic FIR (Finite Impulse Response) equation

\[ y[n] = \sum_{i=0}^{N} b_i x[n - i]. \]

where \( b_i \) is an array of coefficients, and \( x \) the input signal.

Convolution!

In C language

```c
y[n]=0;
For (n=0; n<N; n++)
{
    For (i = 0;i<N;i++)
        y[n] = y[n] + h[i]*x[n-i];
}
```

Only Multiply and Accumulate (MAC) is needed!
Example: Digital Filtering

FIR using General Purpose Processor (GPP)

```
<table>
<thead>
<tr>
<th>Instruction</th>
<th>Description</th>
<th>Symbol(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clr A</td>
<td>Clear Accumulator A</td>
<td>A</td>
</tr>
<tr>
<td>Clr B</td>
<td>Clear Accumulator B</td>
<td>B</td>
</tr>
<tr>
<td>MOV *R0, Y0</td>
<td>Move data from memory location 1 to register Y0</td>
<td>R0, Y0</td>
</tr>
<tr>
<td>MOV *R1, X0</td>
<td>Move data from memory location 2 to register X0</td>
<td>R1, X0</td>
</tr>
<tr>
<td>Mpy X0,Y0,A</td>
<td>X0*Y0 -&gt; A</td>
<td>X0, Y0, A</td>
</tr>
<tr>
<td>Add A,B</td>
<td>A + B -&gt; B</td>
<td>A, B</td>
</tr>
<tr>
<td>Inc R0</td>
<td>R0 + 1 -&gt; R0</td>
<td>R0</td>
</tr>
<tr>
<td>Inc R1</td>
<td>R1 + 1 -&gt; R1</td>
<td>R1</td>
</tr>
<tr>
<td>Dec N</td>
<td>Dec N (initially equals to 3)</td>
<td>N</td>
</tr>
<tr>
<td>Tst N</td>
<td>Test for the value</td>
<td>N</td>
</tr>
<tr>
<td>Jnz Loop</td>
<td>Different than zero loop again</td>
<td>Loop</td>
</tr>
<tr>
<td>Mov B,*R2</td>
<td>Move result to memory</td>
<td>B, R2</td>
</tr>
</tbody>
</table>
```

Real-Time Image Processing - Vibot
Example: Digital Filtering

1. FIR using a Digital Signal Processor (DSP)

```
<table>
<thead>
<tr>
<th>Clr</th>
<th>A</th>
<th>; Clear Accumulator A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rep</td>
<td>N</td>
<td>; Rep N times the next instruction</td>
</tr>
<tr>
<td>MAC</td>
<td>*(R0)+, *(R1)+, A</td>
<td>; Fetch the two memory locations pointed by R0 and R1, multiply them together and add the result to A, the final result is stored back in A</td>
</tr>
<tr>
<td>Mov</td>
<td>A, *R2</td>
<td>; Move result to memory</td>
</tr>
</tbody>
</table>
```

FIR using an FPGA (and Handel-C)

while(1)
{
  par
  {
    /*Pipeline stage 1: get data */
    Input = InputBus.in;
    
    /*Pipeline stage 2 - deal with data */
    
    /* Pipeline stage 3 - perform multiplications */
    par (i = 0; i != TAPS; i++) {
      MultResult[i] = adjs(DataArray[i], (DATAWIDTH * 2)) * CoeffData[i];
    }
    
    /* Pipeline stage 4 - perform additions */
    Output = AddMultResult(TAPS);
  }
}

macro expr AddMultResult(n) =
  select(n == 0, 0, adjs(MultResult[n - 1], ResultWidth) +
  AddMultResult(n - 1));
Example: Digital Filtering

- FIR using an FPGA (and Handel-C)

Stage 1

Stage 2

Stage 3

Stage 4

d[0] to d[8] are the nine items of data

c[0] to c[8] are the nine coefficients

output
Platforms

- GPPs drawbacks
  - More instructions/task
  - Common memory for data and program
    - Limited bus/memory bandwidth

- Solution? DSPs, FPGAs, ASICs...
Platforms

- GPPs

Diagram:
- Memory
- Register 1
- Register 2
- ALU

Memory Data Bus
Platforms

- DSPs

Program Memory Data Bus

Program Memory

Data Memory

Multiplexer

Multiplexer

Data Memory Data Bus

ALU

Accumulator
Memory structures

**Von Neuman architecture**
Area efficient but requires higher bus bandwidth because instructions and data must compete for memory.

**Harvard architecture** was coined to describe machines with separate memories. Speed efficient: Increased parallelism.
DSP versus GPP

- **Multiple parallel units**
  - multiply accumulate (possibly several units)
  - address calculation in parallel to processing
  - barrel shifter

- **Memory Access**
  - special ALU for address calculation
  - Bit reversed addressing
  - circular addressing

- **Automatic loops**
  - Software looping: writing assembly code to perform branching
  - Hardware looping: dedicated hardware loop counter register
DSP Alternatives

Wireless Systems requires more and more high performance and higher bandwidth.

- 2G: 100 MIPS, 8-13 Kbps
- 2.5G: 10,000 MIPS, 64-384 Kbps
- 3G: ~100,000 MIPS, 384-2000 Kbps

DSP performance might not be enough for future applications.
DSP alternatives

- High-performance GPPs with DSP enhancements.
  - Eliminating the need of a DSP and GPP for many products and thus reducing cost
  - Example: Pentium 4
    - Single Instruction Multiple Data (SIMD) instructions allowing identical operations on multiple pieces of data in parallel.
    - 144 new special instructions providing advanced capabilities for applications such as 3D graphics, video encoding/decoding, and speech recognition.
    - Several Data Types (floating/integer)

- Multi-Core DSPs
- Application Specific Integrated Circuits (ASIC)
- Field Programmable Gate Array (FPGA)
ASICS

- Uses hard-wired logic with varied architectures according to the application

- Advantages
  - Speed
  - Reduced Power Consumption
  - Cost/performance
  - Design Flexibility

- Disadvantages
  - Large development costs
  - Lengthy development cycles
  - Inflexibility
FPGAs

- Network of reconfigurable hardware with reconfigurable interconnect controlled by a switching matrix
- Recently includes DSP features
  - ALTERA (e.g.: Stratex) & XILINX (e.g.: Virtex)
- Advantages:
  - More Flexible than ASIC
  - Huge Performance Gain in Some Applications
  - Re-use Hardware for different applications
- Disadvantages:
  - Long Development Cycle
  - Expensive compared to DSP
  - Much higher power consumption compared to DSP
  - Slow time to market compared to DSP

Real-Time Image Processing - Vibot Master - UdG -
FPGAs

- FPGA used in this course. Xilinx Spartan 3 (3S1500)

<table>
<thead>
<tr>
<th>Spartan-3</th>
<th>XC3S50</th>
<th>XC3S200</th>
<th>XC3S400</th>
<th>XC3S1000</th>
<th>XC3S1500</th>
<th>XC3S2000</th>
<th>XC3S4000</th>
<th>XC3S5000</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Gates</td>
<td>50K</td>
<td>200K</td>
<td>400K</td>
<td>1000K</td>
<td>1500K</td>
<td>2000K</td>
<td>4000K</td>
<td>5000K</td>
</tr>
<tr>
<td>Logic Cells</td>
<td>1,728</td>
<td>4,320</td>
<td>8,064</td>
<td>17,280</td>
<td>29,952</td>
<td>46,080</td>
<td>62,208</td>
<td>74,880</td>
</tr>
<tr>
<td>18x18 Multipliers</td>
<td>4</td>
<td>12</td>
<td>16</td>
<td>24</td>
<td>32</td>
<td>40</td>
<td>96</td>
<td>104</td>
</tr>
<tr>
<td>Block RAM Bits</td>
<td>72K</td>
<td>216K</td>
<td>288K</td>
<td>432K</td>
<td>576K</td>
<td>720K</td>
<td>1,728K</td>
<td>1,872K</td>
</tr>
<tr>
<td>Distributed RAM Bits</td>
<td>12K</td>
<td>30K</td>
<td>56K</td>
<td>120K</td>
<td>208K</td>
<td>320K</td>
<td>432K</td>
<td>520K</td>
</tr>
<tr>
<td>DCMs</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>I/O Standards</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>Max Differential I/O Pairs</td>
<td>56</td>
<td>76</td>
<td>116</td>
<td>175</td>
<td>221</td>
<td>270</td>
<td>312</td>
<td>344</td>
</tr>
<tr>
<td>Max Single Ended I/O</td>
<td>124</td>
<td>173</td>
<td>264</td>
<td>391</td>
<td>487</td>
<td>565</td>
<td>712</td>
<td>784</td>
</tr>
</tbody>
</table>
FPGAs

- A FPGA consists of programmable logic blocks and routing
- In most FPGAs, the basic logic block contains a Lookup Table (LUT) and a flip-flop (FF)
- Logic is implemented in these LUTs
- There may well be other logic, such as multiplexers and carry logic
- Fast and configurable routing resources.
- There are also dedicated low-skew clock nets
- Modern FPGAs also have other important resources
  - Dedicated memory resources
  - Dedicated arithmetic blocks (ALUs)
  - Processor cores
  - Fast serial I/O

Real-Time Image Processing - Vibot Master - UdG -
**FPGAs**

- **Xilinx Spartan 3**
  - **Configurable Logic Blocks (CLBs).** RAM-based Look-Up Tables (LUTs) to implement logic and storage elements that can be used as flip-flops or latches.
  - **Input/Output Blocks (IOBs)** control the flow of data between the I/O pins and the internal logic of the device. Each IOB supports bidirectional data flow plus 3-state operation. Block RAM provides data storage in the form of 18-Kbit dual-port blocks.
  - **Multiplier blocks** accept two 18-bit binary numbers as inputs and calculate the product.
  - **Digital Clock Manager (DCM) blocks** provide self-calibrating, fully digital solutions for distributing, delaying, multiplying, dividing, and phase shifting clock signals.
FPGAs

- Xilinx Spartan 3
FPGAs

- Xilinx Spartan 3
FPGAs

How big is a FPGA? For RTI not enough!

- People usually talk about “gates” or “ASIC equivalent gates”
- The only absolute figures you can use are
  - The number of LUT/FF pairs
  - CLBs/Slices in Xilinx
  - LABs/LEs in Altera
  - The number of RAM bits
  - The number of hard arithmetic units

For exact values, refer to the device data book!

XILINXfPGAds099.pdf
FPGAs

- How big is a FPGA? For RTI not enough!
  - People usually talk about “gates” or “ASIC equivalent gates”
  - The only absolute figures you can use are
    - The number of LUT/FF pairs
    - CLBs/Slices in Xilinx
    - LABs/LEs in Altera
    - The number of RAM bits
  - For XILINX

<table>
<thead>
<tr>
<th>Spartan-3</th>
<th>XC 3S50</th>
<th>XC 3S200</th>
<th>XC 3S400</th>
<th>XC 3S1000</th>
<th>XC 3S1500</th>
<th>XC 3S2000</th>
<th>XC 3S4000</th>
<th>XC 3S5000</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Gates</td>
<td>50K</td>
<td>200K</td>
<td>400K</td>
<td>1000K</td>
<td>1500K</td>
<td>2000K</td>
<td>4000K</td>
<td>5000K</td>
</tr>
<tr>
<td>Logic Cells</td>
<td>1,728</td>
<td>4,320</td>
<td>8,064</td>
<td>17,280</td>
<td>29,952</td>
<td>46,080</td>
<td>62,208</td>
<td>74,880</td>
</tr>
<tr>
<td>18x18 Multipliers</td>
<td>4</td>
<td>12</td>
<td>16</td>
<td>24</td>
<td>32</td>
<td>40</td>
<td>96</td>
<td>104</td>
</tr>
<tr>
<td>Block RAM Bits</td>
<td>72K</td>
<td>216K</td>
<td>288K</td>
<td>432K</td>
<td>576K</td>
<td>720K</td>
<td>1,728K</td>
<td>1,872K</td>
</tr>
<tr>
<td>Distributed RAM Bits</td>
<td>12K</td>
<td>30K</td>
<td>56K</td>
<td>120K</td>
<td>200K</td>
<td>320K</td>
<td>432K</td>
<td>520K</td>
</tr>
<tr>
<td>DCMs</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>I/O Standards</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>Max Differential I/O Pairs</td>
<td>56</td>
<td>76</td>
<td>116</td>
<td>175</td>
<td>221</td>
<td>270</td>
<td>312</td>
<td>344</td>
</tr>
<tr>
<td>Max Single Ended I/O</td>
<td>124</td>
<td>173</td>
<td>264</td>
<td>391</td>
<td>487</td>
<td>565</td>
<td>712</td>
<td>784</td>
</tr>
</tbody>
</table>
FPGAs

- LUTs
  - Both Altera and Xilinx FPGAs have 4-input LUTs (except Stratix II)
  - A LUT can be thought of as implementing the truth table for a four input, one output circuit
  - Any circuit with four inputs and one output can be mapped into a single LUT
  - Each LUT’s truth table is fixed at start-up, so a LUT only ever performs the same logic function
  - A different logic function with the same four inputs, another LUT will be used
LUTs. Example.

```c
unsigned 1 a, b, c, d, e;
e = (a ^ b) | (c & d);
```

If the code has 5 inputs, it maps to 2 LUTs!
FPGAs

- **Place & Route**
  - Placement is the process of putting individual components in specific locations in the device.
  - Routing is the process of connecting all the components together in the device such that the routing delay between components meets timing constraints.

- Always check timing report.
- Look for changes in mapping report.
  - Is something being optimised away?
- Learn how to set effort levels.
FPGAs

Place & Route. Example in DK

Device Utilization Summary:

- Number of BUFGs: 5 out of 8 - 62%
- Number of DCMs: 2 out of 4 - 50%
- Number of External IOBs: 60 out of 221 - 27%
- Number of LOCed IOBs: 50 out of 60 - 100%
- Number of RAMB16s: 8 out of 92 - 90%
- Number of Slices: 876,035 / 1,331,2 - 6%
- Number of SLICEMs: 104 out of 6,666 - 1%

Overall effort level (-ol): Standard
Placer effort level (-pl): High
Placer cost table entry (-t): 1
Router effort level (-rl): Standard

Starting initial Timing Analysis. REAL time: 7 secs
Finished initial Timing Analysis. REAL time: 7 secs
Starting Placer
Phase 1.1
Phase 1.1 (Checksum: 98bfe7) REAL time: 9 secs

Starting Router
Phase 1: 5968 unrouted; REAL time: 37 secs
Phase 2: 4886 unrouted; REAL time: 35 secs
Phase 3: 943 un routed; REAL time: 20 secs
Phase 4: 943 unrouted; REAL time: 41 secs
Phase 5: 1024 unrouted; 4 mins 39: REAL time: 445474

Real-Time Image Processing - Vibot Master - UdG -
FPGAs

Will my design fit?

“How many NAND gates will fit in my FPGA?” It depends!

No direct correspondence between NAND gates and LUTs

The LUT, Flip-Flop and memory counts are the significant numbers

If it fits in terms of LUT and flip-flops, it may not place or route

There may not be enough routing resources

e.g. if you are using 99% of FFs and LUTs, the tools might not be able to route your design

Conclusion: place and route your design before making any firm conclusions.
To know more...

RTI in Hardware

Robert Martí