Reconfigurable Interconnect for Next Generation Systems

Ingrid Verbauwhede, M.F. Chang
Electrical Engineering Department, UCLA
7440B Boelter Hall
ingrid@ee.ucla.edu

with thanks to: Jongsun Kim, Patrick Schaumont, UCLA
Outline

- Post PC Era – Heterogeneous distributed embedded systems
- Challenge 1: Energy – Flexibility conflict
  - Approach: Architecture design
    - Domain specialization
    - Reconfiguration hierarchy
- Challenge 2: Interconnect & Memory bandwidth
  - Approach:
    - Reconfigurable RF – Interconnect
      - On chip
      - Off chip
  - Impact to SLIP
- Conclusions
Next generation applications

Sensor based poetic/informational interactive system
B. Seaman,
(Digital Media Arts)
I. Verbauwhede

Macromedia Shockwave: http://students.dma.ucla.edu/~fwinkler/Poly-Sensing

Ingrid Verbauwhede
Challenge 1: Energy – Flexibility trade-off
The Energy-Flexibility Conflict

Computing efficiency (GOPS/Watt)

- High end MPEG - 4
- Low end CIF decoding
- DSP-VLIW
- ISProcessors
- microprocessors

[ T.Claasen et al. ISSCC99]
### Example: AES co-processor

<table>
<thead>
<tr>
<th>AES 128bit key 128bit data</th>
<th>Throughput</th>
<th>Power</th>
<th>Figure of Merit (Gb/s/W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain Specific Processor (0.18 μm CMOS) [1]</td>
<td>1.28 Gbits/sec</td>
<td>56 mW</td>
<td>22.8 (100%)</td>
</tr>
<tr>
<td>FPGA [2],[3]</td>
<td>640 Mbits/sec</td>
<td>1.63 W</td>
<td>0.39 (1.7%)</td>
</tr>
<tr>
<td>Pentium III [4],[5]</td>
<td>607 Mbits/sec</td>
<td>41.4 W</td>
<td>0.015 (0.06%)</td>
</tr>
</tbody>
</table>

[2] Altera Technical Brief TB57: Xilinx VII 700 mW @ 100 MHz, 660 CLB
[3] 3th AES Candidate Conference, Elbirt et al: Xilinx VII 14.1 MHz, 5302 CLB (w/o key schedule = overhead 33%)
[5] Intel Pentium III Datasheet 1.13 GHz (Vcc=1.8V, Icc=23A)
Approach: domain specialization

- General purpose processor is not the solution
  (programmable at instruction set)
- FPGA is not the solution
  ("programmable" = reconfigurable at CLB level)
- Fixed ASIC is not the solution

Reconfiguration hierarchy (architecture view)
& domain specialization (system view)

- Domains: wireless communications, security, network processing
  (high throughput, low energy, unusual arithmetic)
Domain specific processing

General Purpose Application

Narrow the Mapping Gap

General Purpose Platform

Domain Specific Application

Cover lots of Applications

Domain Specific Platform

Non-programmable ASIC fixed point
Trading on the Energy-Flexibility Boundary is hard

Domain-Specialization

Specialized Micro-Architectures

FPGA

Xilinx

Altera

Actel

Adaptive Silicon

Specialized Instruction-Set Architectures

Processor

Morphics

PMC Sierra

Improv Systems

Adelante

Target Compiler

ARC

Tensilica

[DAC2001]
Reconfiguration Hierarchy
= 3D Design Space

<table>
<thead>
<tr>
<th>Reconfigurable Feature</th>
<th>Computation Abstraction Level</th>
<th>Binding Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Communication</th>
<th>Storage</th>
<th>Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implementation</td>
<td>Switches</td>
<td>RAM Organization</td>
<td>CLB</td>
</tr>
<tr>
<td></td>
<td>Muxes</td>
<td></td>
<td>Parametrizable IP-block</td>
</tr>
<tr>
<td>Micro-Architecture</td>
<td>Crossbar</td>
<td>Register File Size</td>
<td>Execution Unit Type</td>
</tr>
<tr>
<td></td>
<td>Busses</td>
<td>Cache Architecture</td>
<td>Interpreter Levels</td>
</tr>
<tr>
<td>Instruction Set</td>
<td>Size of address/data bus</td>
<td>Register Set</td>
<td>Custom Instructions</td>
</tr>
<tr>
<td>Architecture</td>
<td></td>
<td>Memory Architecture</td>
<td>Interrupt Architecture</td>
</tr>
<tr>
<td>Process Architecture/Systems Architecture</td>
<td>Interconnection network</td>
<td>Buffer Size</td>
<td>Number and type of asynchronous processes and tasks</td>
</tr>
</tbody>
</table>
Challenge 2: reconfigurable interconnect

Datapaths are NOT the problem
but
Memory bandwidth and interconnect are!
DSP Processor Fundamentals

Processor Components [Skillikorn-88]

Adapt ALL components to the application domain!
**Example: FIR implementation**

\[ y(n) = \sum_{i=0}^{N-1} c(i) x(n-i) \]

\[ y(0) = c(0)x(0) + c(1)x(-1) + c(2)x(-2) + \ldots + c(N-1)x(1-N); \]
\[ y(1) = c(0)x(1) + c(1)x(0) + c(2)x(-1) + \ldots + c(N-1)x(2-N); \]
\[ y(2) = c(0)x(2) + c(1)x(1) + c(2)x(0) + \ldots + c(N-1)x(3-N); \]
\[ \ldots \]
\[ y(n) = c(0)x(n) + c(1)x(n-1) + c(2)x(n-2) + \ldots + c(N-1)x(n-(N-1)); \]

Software (Von Neumann): Execute row by row = 2 reads for one MAC
FIR on DSP Lode™ (1996)

- FIR filter: two outputs in parallel with delay register

\[
\begin{align*}
y(0) &= c(0)x(0) + c(1)x(-1) + c(2)x(-2) + \ldots + c(N-1)x(1-N); \\
y(1) &= c(0)x(1) + c(1)x(0) + c(2)x(-1) + \ldots + c(N-1)x(2-N); \\
y(2) &= c(0)x(2) + c(1)x(1) + c(2)x(0) + \ldots + c(N-1)x(3-N); \\
y(3) &= c(0)x(3) + c(1)x(2) + c(2)x(1) + \ldots + c(N-1)x(4-N);
\end{align*}
\]

- Two MAC units with dedicated bus network

2 reads for 2 MAC’s
(or 2 reads for N MAC’s)
ONLY 2 busses!
FIR on TI C55x (2000)

- FIR filter: two outputs in parallel with 3 busses

\[
\begin{align*}
y(0) &= c(0)x(0) + c(1)x(-1) + c(2)x(-2) + \ldots + c(N-1)x(1-N); \\
y(1) &= c(0)x(1) + c(1)x(0) + c(2)x(-1) + \ldots + c(N-1)x(2-N); \\
y(2) &= c(0)x(2) + c(1)x(1) + c(2)x(0) + \ldots + c(N-1)x(3-N); \\
y(3) &= c(0)x(3) + c(1)x(2) + c(2)x(1) + \ldots + c(N-1)x(4-N); 
\end{align*}
\]
## Energy comparison

### Total energy for one output sample:

<table>
<thead>
<tr>
<th>Energy</th>
<th>Single MAC</th>
<th>Dual MAC</th>
<th>Dual MAC 3 busses</th>
<th>Dual MAC with REG</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of MAC operations</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>No of Memory reads</td>
<td>2N</td>
<td>2N</td>
<td>1.5N</td>
<td>N</td>
</tr>
<tr>
<td>No of Instruction Cycles</td>
<td>N</td>
<td>N/2</td>
<td>N/2</td>
<td>N/2</td>
</tr>
</tbody>
</table>

Adaptation of the datapath: MAC, DMAC
Adaptation of the memory architecture and bus network
Adaptation of the instruction set
Example 2: Viterbi

- Viterbi butterfly
  - $i =$ state index
  - $s =$ # of states = $2^{k-1}$
  - $w =$ decoding window

- Basic equations:
  - $d(2i) = \min \{ d(i) + a, d(i + s/2) - a \}$
  - $d(2i + 1) = \min \{ d(i) - a, d(i + s/2) + a \}$

- Key operation: Add-Compare-Select (ACS)

- IS-95: $k = 9$, 256 states, $w = 192$, means $2^8 \times 192 \times$ (cycles for one ACS)
Two MAC units & ALU: Add-Compare-Select

\[ \Gamma = \min [(\Gamma_1 + \mu_1), (\Gamma_2 + \mu_2)] \]

- DMAC operates as dual add/subtract unit
- ALU finds minimum
- Shortest distance saved
- Path indicator saved
- 4 cycles / butterfly
Viterbi on TIC54x

- ALU and CSSU: CMPS instruction

\[ \Gamma = \min [(\Gamma_1 + \mu_1), (\Gamma_2 + \mu_2)] \]

- ALU splits in 16 bit halves
- ACC splits in half
- Shortest distance saved
- CSSU compares halves
- Path indicator saved
- 4 cycles / butterfly

Source: TI Application Report, Viterbi Decoding in the TMS320C54x family, document SPRA071

SLIP, San Diego, April 6, 2002
Ingrid Verbauwhede
Viterbi on TI 'C6x

3-cycle 2-ACS Inner-Loop

<table>
<thead>
<tr>
<th>LOOP:</th>
<th>b .s1 LOOP</th>
<th>[b1] sub .s2 b1,1,b1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[a2] sth .d1 b12,*a6[8]</td>
<td>[b1] cmpgt .l1 a11,a10,a1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[a2] add .d2 b0,b14,b14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>cmpgt .l2 b11,b10,b0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>mpy .m1x 1,b5,a4</td>
</tr>
<tr>
<td>[a2]</td>
<td>sub .s1 a2,1,a2</td>
<td></td>
</tr>
<tr>
<td>[a2]</td>
<td>sth .d1 a12,*a6++</td>
<td></td>
</tr>
<tr>
<td>[a1]</td>
<td>add .s2 2,b0,b0</td>
<td></td>
</tr>
<tr>
<td>[b0]</td>
<td>mpy .m2 1,b11,b12</td>
<td></td>
</tr>
<tr>
<td></td>
<td>mpy .mlx 1,b10,b12</td>
<td></td>
</tr>
<tr>
<td></td>
<td>sub .d2x a7,b5,b10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ldh .d2 *++b9,b5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>shl .s2 b14,2,b14</td>
<td></td>
</tr>
<tr>
<td>[a1]</td>
<td>mpy .mlx 1,a11,a12</td>
<td></td>
</tr>
<tr>
<td></td>
<td>add .s1 a7,a4,a10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>sub .d1x b13,a4,a11</td>
<td></td>
</tr>
<tr>
<td></td>
<td>add .d2 b13,b5,b11</td>
<td></td>
</tr>
<tr>
<td></td>
<td>mpy .m2 1,b10,b12</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ldh .d2 *b4++[2],a7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ldh .d1 *a5++[2],b13</td>
<td></td>
</tr>
</tbody>
</table>

; end of LOOP

- 16-state Viterbi decoder for GSM
- 3 cycles per butterfly
- 32 cycles per GSM timeslot (8 butterflies)
- MPY instructions used to move data!

---

[ISSCC 2000 Tutorial] Ingrid Verbauwhede
Reconfigurable Interconnect

- Is expensive!
- FPGA power breakdown:

  - Interconnect: 65%
  - Clock: 21%
  - I/O: 9%
  - CLB: 5%

[Source: E. Kusse, ISLPED 1998]
Reconfigurable interconnect

“We don’t know what the customer will run”

World’s Fastest Logic & Routing

Conexant 3.125Gb Serial
IBM PowerPC® RISC CPU
XtremeDSP™
Synchronous Dual-Port RAM
SelectIO-Itra™ SystemIO™ & XCITE™

FPGA 2002: dynamic power: 60% routing, 16% logic, 14% clock!

SLIP, San Diego, April 6, 2002
Ingrid Verbauwhede
Xilinx – Off Chip

“We don’t know what the customer will run”

System-Synchronous
Parallel
- PCI-66
- PCI-X133

Source-Synchronous
Parallel
- RapidIO
- HyperTransport
- SPI-4
- POS-PHY3
- POS-PHY4
- FlexBus3
- FlexBus4
- XGMII

Advanced
Memory Interfaces
- ZBT SSRAM
- DDR SDRAM
- QDR SSRAM
- CAM

High-Speed Serial
- Multi-Gigabit
Backplanes
- Infiniband™
- Gb Ethernet
- 10 Gb Ethernet (XAUI)
- Fibre Channel

[courtesy: Xilinx]
Approach: RF-Interconnect

Conventional interconnect:
- space division
- time division

New interconnect:
- code division (CDMA)
- frequency division (FDMA)
- any combination of the above
  
  Reconfigurable!

- “Low Loss, dispersion-free, ultra-high data rate (100Gbps/channel & 20Tbps/chip)”

[M.F. Chang, Proc. IEEE 2001]
CDMA-Interconnect

Shared Transmission Line
\[ s(t) = U_1W_1 + U_2W_2 + \ldots + U_MW_n \]

Data for User 1
\( W_{1/M}(t) \)
- Spreading
- Coupling Network
- Despreading

Data from User 1/M
\( f_{\text{baud}} \)
\( f_{\text{chip}} \)
\( W_{1/M}(t) \)

Data for User M
\( W_{M/1}(t) \)
- Spreading
- Coupling Network
- Despreading

Data from User M/1
\( f_{\text{baud}} \)
\( f_{\text{chip}} \)
\( W_{M/1}(t) \)

[M.F. Chang, Proc. IEEE 2001]

[Image of diagram]
Channel Reconfiguration at $f_{clk}=2.8\text{GHz}$

- **Reconfiguration Enable**
- **Tx-1**
- **Tx-2**
- **Rx-1**
- **Rx-2**

Transition Time $< T_S$
RF – Off chip: CDMA based SDRAM bus

- Only 1 PCB line for 8 bit data transfer
- PCB line is terminated at both end of the line.
- Small Swing Multi-Level Signaling
- Source Synchronous Clocking
  (Clock and Data Transfer together)

[courtesy: Jongsun Kim, UCLA]
SDRAM interface: conventional bus

Master (CPU) -- Physical Channel -- Slave (DRAM)

- # of 8 PCB lines for 8 bit data transfer
- PCB line is terminated (Rambus, DDR)
- Source Synchronous Clocking (Rambus, DDR)

[courtesy: Jongsun Kim, UCLA]
Comparison

<table>
<thead>
<tr>
<th></th>
<th># of Parallel high-speed Data Channels</th>
<th># of Parallel High-speed Address Channels</th>
<th># of Parallel High-speed Clock Channels</th>
<th>Total # of Parallel High-speed Channels</th>
<th>Min. # of Shielding Channels for Data &amp; Add.</th>
<th>Data Rate (Mbps)</th>
<th>Total Data Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDRAM (PC-133)</td>
<td>64</td>
<td>12</td>
<td>1</td>
<td>77</td>
<td>-</td>
<td>133</td>
<td>1.1 Gbyte/s</td>
</tr>
<tr>
<td>Rambus DRAM</td>
<td>16</td>
<td>8</td>
<td>(differential)</td>
<td>28</td>
<td>24</td>
<td>400 * 2</td>
<td>1.6 Gbyte/s</td>
</tr>
<tr>
<td>DDR(SSTL-2)</td>
<td>64</td>
<td>12</td>
<td>(data strobe)</td>
<td>84</td>
<td>-</td>
<td>133 * 2</td>
<td>2.1 Gbyte/s</td>
</tr>
<tr>
<td>CDMA DRAM</td>
<td>2</td>
<td>1</td>
<td>(differential)</td>
<td>2</td>
<td>3</td>
<td>1600 * 4</td>
<td>1.6 Gbyte/s</td>
</tr>
</tbody>
</table>

- RDRAM use 2 differential (CTM/CTMB,CFM/CFMB) clock lines for source synchronous clocking
- DDR use 8 data strobe (DQS) lines for source synchronous clocking
- CDMA DRAM use 2 differential clock lines for source synchronous clocking
- Both SDRAM and DDR DRAM has some additional command lines which are not listed above.

[courtesy: Jongsun Kim, UCLA]
## CDMA DRAM Bandwidth options

<table>
<thead>
<tr>
<th></th>
<th>Rambus DRAM</th>
<th>CDMA DRAM case 1</th>
<th>CDMA DRAM case 2</th>
<th>CDMA DRAM case 3</th>
<th>CDMA DRAM case 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRAM Core Freq.</td>
<td>100MHz</td>
<td>100MHz</td>
<td>100MHz</td>
<td>100MHz</td>
<td>100MHz</td>
</tr>
<tr>
<td>DRAM Core Internal Bus</td>
<td>8bit*C=128b</td>
<td>8bit<em>N</em>C=128b</td>
<td>8bit<em>N</em>C=1024b</td>
<td>8bit<em>N</em>C=128b</td>
<td>8<em>N</em>C=512b</td>
</tr>
<tr>
<td>DRAM Interface Freq.</td>
<td>400 MHz * 2</td>
<td>400 MHz * 2</td>
<td>400MHz * 2</td>
<td>300MHz * 2</td>
<td>300MHz * 2</td>
</tr>
<tr>
<td>DLL Freq.</td>
<td>400 MHz</td>
<td>1.6 GHz</td>
<td>1.6 GHz</td>
<td>1.2 GHz</td>
<td>1.2 GHz</td>
</tr>
<tr>
<td>I/O Interface Freq.</td>
<td>400 MHz * 2</td>
<td>1.6 GHz * 2</td>
<td>1.6 GHz * 2</td>
<td>1.2 GHz * 2</td>
<td>1.2 GHz * 2</td>
</tr>
<tr>
<td># of Data PCB Channels</td>
<td>C=16</td>
<td>C=2</td>
<td>C=16</td>
<td>C=4</td>
<td>C=16</td>
</tr>
<tr>
<td># of CDMA Users</td>
<td>-</td>
<td>N=8</td>
<td>N=8</td>
<td>N=4</td>
<td>N=4</td>
</tr>
<tr>
<td>Total Bandwidth</td>
<td>1.6 Gbyte/s</td>
<td>1.6 Gbyte/s</td>
<td>12.8 Gbyte/s</td>
<td>1.6 Gbyte/s</td>
<td>6.4 Gbyte/s</td>
</tr>
</tbody>
</table>

- Rambus DRAM use dual edge of 400MHz DLL clock for I/O Interface
- CDMA DRAM use dual edge of 1.6GHz DLL clock for I/O CDMA Interface

[courtesy: Jongsun Kim, UCLA]
Andrew’s question: what is the message for SLIP?
- Max capacity for min energy

Example: EE215B (Advanced Digital Integrated Circuits Class)
- 10” lossy transmission line, 1 Gbs
- Study termination and coupling noise
- Part II: Inter symbol interference,
  - max throughput for given length (10”)
  - Max length for given throughput (1Gbs)

Part III: low swing
Basic Termination Methods

- **Series termination**
  - $V_{in}(t)$
  - $Z_0$

- **Parallel termination**
  - $V_{in}(t)$
  - $Z_0$

- **Split parallel(Thevenin) termination**
  - $V_{in}(t)$
  - $Z_0$
  - $R_{T1} || R_{T2} = Z_0$

- **AC parallel termination**
  - $V_{in}(t)$
  - $Z_0$
  - $C_T$

- **Series + Schottky Diode termination**
  - $V_{in}(t)$
  - $Z_0$

- **Series + Parallel termination**
  - $V_{in}(t)$
  - $Z_0$
  - (Gnd or Vdd)
Received node voltage probing I

<table>
<thead>
<tr>
<th>Wave</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>D0</td>
<td>AC</td>
</tr>
<tr>
<td>D1</td>
<td>Parallel + Gnd</td>
</tr>
<tr>
<td>D2</td>
<td>Parallel + Vdd</td>
</tr>
<tr>
<td>D3</td>
<td>Series</td>
</tr>
<tr>
<td>D4</td>
<td>Split Parallel</td>
</tr>
</tbody>
</table>

Input

Graph showing voltage waveform for different nodes:
- D0 (AC)
- D1 (Parallel + Gnd)
- D2 (Parallel + Vdd)
- D3 (Series)
- D4 (Split Parallel)

Legend:
- Parallel + Vdd
- Split Parallel
- Series
- AC
- Parallel + Gnd
Advanced Termination Methods

- **GTL**

- **RSL**

- **SSTL-2**
Inter Symbol Interference

1. Length=30 inches and increase data-rate until 8.3 Gbps

(consider ideal signal level (no ISI): High=750mV, Low=0mV)

![Graph showing output distribution](image)

- Positive difference
- Negative difference

Series + parallel termination
ISI: Eye Diagram

Length=30 inches and increase data-rate until 8.3 Gbps

Next step: multilevel signals
Optimal voltage swing

Trade-off
- power of transmitter and receiver
- power into line
10Gbs: min swing 0.16V
Next generation system on a chip

Multi chip module (MCM)

Radio frontend (e.g. SiGe)

Off chip DRAM

Flash Memory

CMOS System on a chip

CPU

Wireless BB Processor

Encryption processor

Memory

Network Processor

ADC/DAC

reconfigurable RF interconnect at MCM level

reconfigurable RF interconnect at IC level
Conclusion

- Challenge 1: energy – flexibility tradeoff
- Goal: max capacity for min energy
- Approach: domain specific processing - reconfiguration hierarchy
- Challenge 2: reconfigurable interconnect
- Approach at the system architecture level
- Approach at “circuit” level: RF interconnect
- Many problems to be addressed