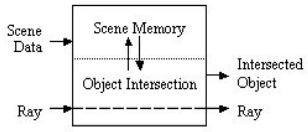
CprE / ComS 583 Reconfigurable Computing

Prof. Joseph Zambreno
Department of Electrical and Computer Engineering
Iowa State University

Lecture #15 – Midterm Review

Project Proposals

- Group 01 – Real-time Ray-Tracing Scene Manager Architecture
- Scene data management
- Ray intersection calculation

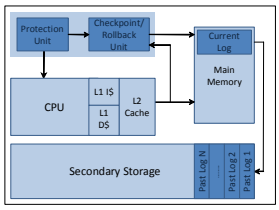


Project Proposals (cont.)

- Group 02 – FPGA-based Emulation of an 8051 Microprocessor
- Memory controller
- Serial interface, ethernet interface

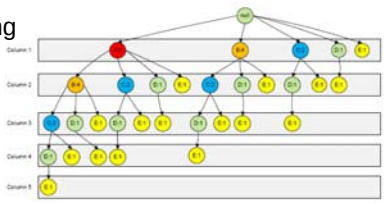
Project Proposals (cont.)

- Group 04 – A Prototype for Verifying Application Integrity and Checkpointing Execution
- Attack detection scheme
- Hardware-accelerated checkpointing scheme



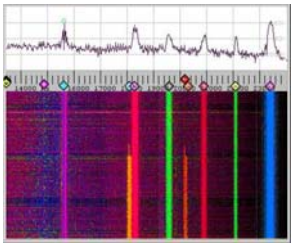
Project Proposals (cont.)

- Group 05 – Implementation of the FP-Growth Algorithm using Reconfigurable Hardware
- Hardware implementation of systolic tree structure
- Benchmarking



Project Proposals (cont.)

- Group 06 – Real-time Audio Spectrography using FPGA
- High-speed FFT implementations
- FPGA interfacing with analog data

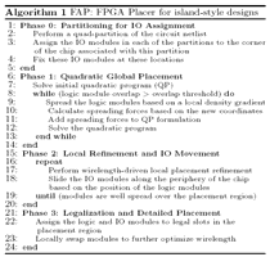


Project Proposals (cont.)

- Group 07 – True Random Bitstream Generator
- Interface with true random noise stream
- Real-time statistical analysis, bias correction

Project Proposals (cont.)

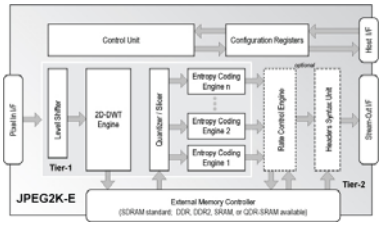
- Group 08 – FAP: A Fast Analytical Placer for Large-Scale FPGA Designs
- Four-phase algorithm integration within VPR
- Benchmarking with real circuit designs


Algorithm 1 FAP: FPGA Placer for island-style designs

1: **Phase 0: Partitioning for IO Assignment**
2: Perform a quad-partition of the circuit netlist
3: Assign the IO modules in each of the partitions to the corner of the chip associated with this partition
4: Fix these IO modules at these locations
5: **end**
6: **Phase 1: Quadratic Global Placement**
7: Solve initial quadratic program (QP)
8: **while** (logic module overlap > overlap threshold) **do**
9: Spread the logic modules based on a local density gradient
10: Calculate spreading forces based on the new coordinates
11: Add spreading forces to QP formulation
12: Solve the quadratic program
13: **end while**
14: **end**
15: **Phase 2: Local Refinement and IO Movement**
16: **repeat**
17: Perform wirelength-driven local placement refinement
18: Slide the IO modules along the periphery of the chip based on the position of the logic modules
19: **until** (modules are well spread over the placement region)
20: **end**
21: **Phase 3: Legalization and Detailed Placement**
22: Assign the logic and IO modules to legal slots in the placement region
23: Locally swap modules to further optimize wirelength
24: **end**

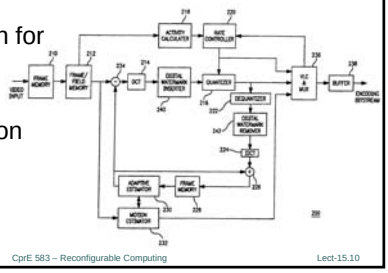
Project Proposals (cont.)

- Group 09 – Implementing the 2-D Wavelet Transform over Reconfigurable Platforms
- Wavelet function selection
- Fixed-point quantization analysis



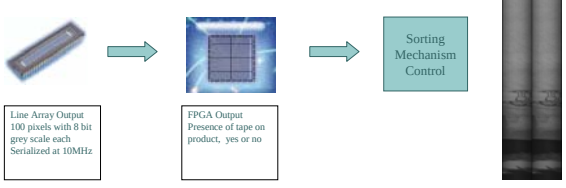
Project Proposals (cont.)

- Group 10 – Video Watermarking System for SD MPEG-2
- Software implementation for comparison
- DCT/iDCT computations on FPGA



Project Proposals (cont.)

- Group 12 – Web Line-Array Sensor
- Conveyor-line control
- Computer vision techniques

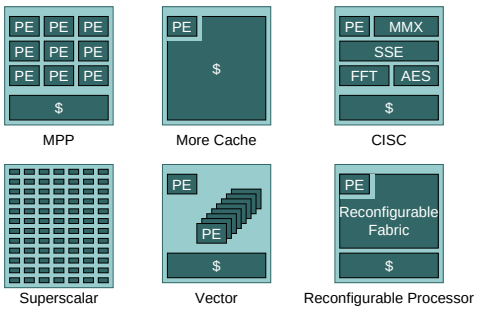


Project Proposals (cont.)

- Reminders:
  - 11/15 – Project Updates (10 minutes)
  - 12/4-12/6 – Final Presentations (25 minutes)
  - 12/14 – Final Reports

Midterm Review

- Using the Silicon

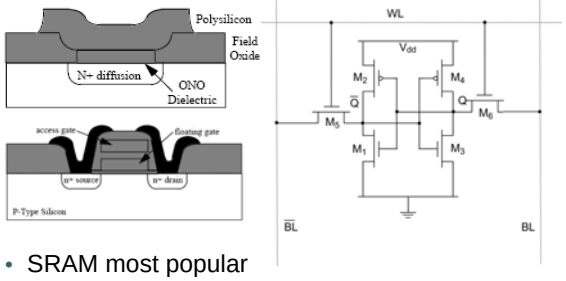


What is Reconfigurable Computing?

- In its current usage, the term reconfigurable computing refers to some form of *hardware programmability*
  - Hardware that can be customized using some physical control points
  - Goal: to *adapt* at the logic level to solve *specific* problems
- Some other definitions:
  - (1) systems incorporating some form of hardware programmability – customizing how the hardware is used using a number of physical control points [Compton, 2002]
  - (2) computing via a post-fabrication and spatially programmed connection of processing elements [Wawrzynek, 2004]
  - (3) general-purpose custom hardware [Goldstein, 1998]

FPGA Technology

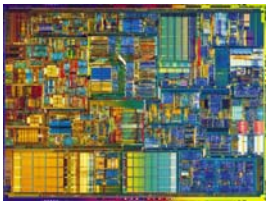
- Various FPGA programming technologies (Anti-fuse, (E)EPROM, Flash, SRAM):



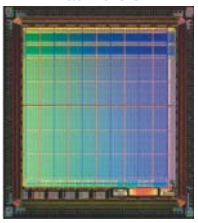
- SRAM most popular

Computational Density (Qualitative)

Intel Pentium 4



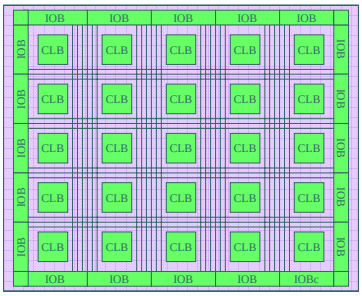
Actel ProASIC



- FPGAs can complete more work per unit time than a processor or DSP:
  - Less instruction overhead
  - More active computation onto the same silicon area (allows for more parallelism)
  - Can control operations at the bit level (as opposed to word level)

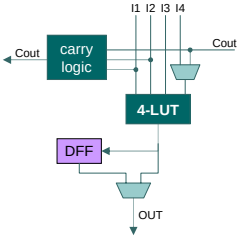
Generic FPGA Architecture

- **FPGA = F**ield-**P**rogrammable **G**ate **A**rray
  - Input/Output Buffers (IOBs)
  - Configurable Logic Blocks (CLBs)
  - Programmable interconnect mesh



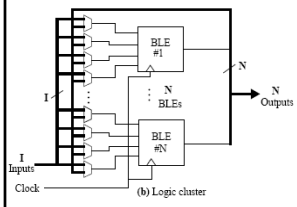
Island-style FPGA architecture

LUT-based Logic Element



- Each LUT operates on four one-bit inputs
- Output is one data bit
- Can perform <u>any</u> Boolean function of four inputs
- 2^{2^4} = 65536 functions (4096 patterns)
- The basic logic element can be more complex
- *Coarse* v. *Fine* grained
- Contains some sort of programmable interconnect

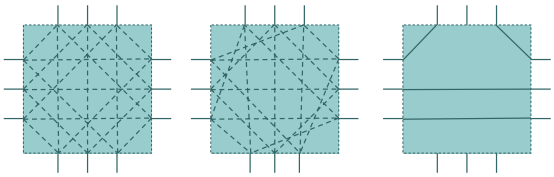
Architectural Issues [AhmRos04A]



- What values of N, I, and K minimize the following parameters?
  - Area
  - Delay
  - Area-delay product
- Assumptions
  - All routing wires length 4
  - Fully populated IMUX
  - Wiring is half pass transistor, half tri-state

Switch Boxes

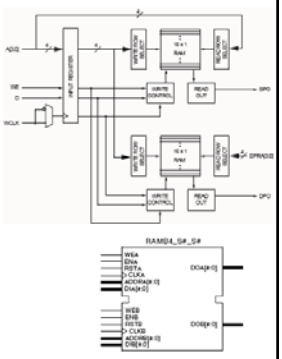
- F_s – connections offered per incoming wire
- *Universal* switchbox can connect any set of inputs to their target output channels simultaneously
  - Build-able with $F_s = 3$
  - Xilinx XC4000 switchbox is $F_s = 3$ but not universal



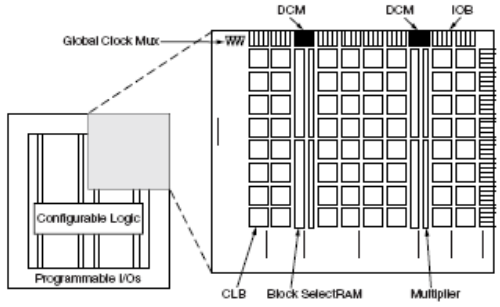
- Read [ChaWon96A] for more details

FPGA Memory Resources

- Individual LUTs can be programmed as 16x1 RAMs and combined to form larger memory structures
- Block Select+RAM – dedicated blocks of on-chip, true dual port read/write synchronous RAM
  - 4Kb (later 18Kb, 36Kb) of RAM with different aspect ratios
  - Faster, less flexible than distributed RAM using LUTs

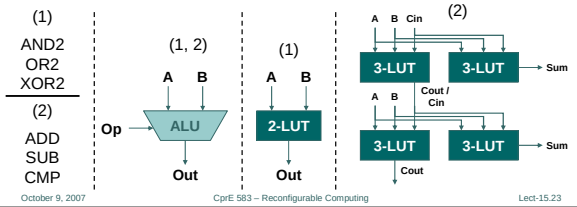


Example FPGA: Xilinx Virtex-II



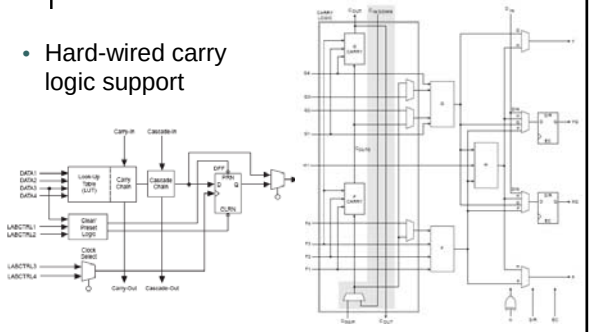
FPGA Arithmetic

- Traditional microprocessors, DSPs, etc. don't use LUTs
- Instead use a *w*-bit Arithmetic and Logic Unit (ALU)
  - Carry connections are hard-wired
  - No switches, no stubs, short wires



FPGA Arithmetic (cont.)

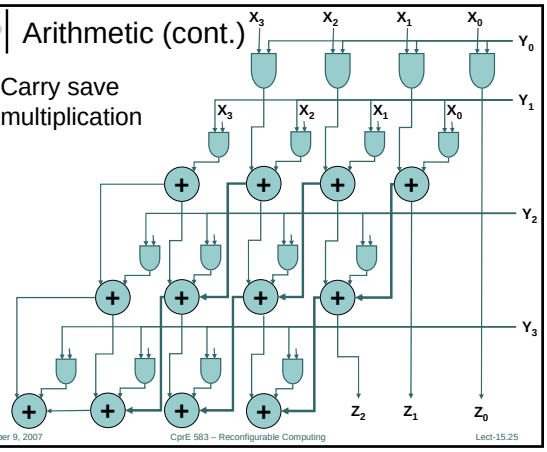
- Hard-wired carry logic support



Altera FLEX 8000 Xilinx XCV4000

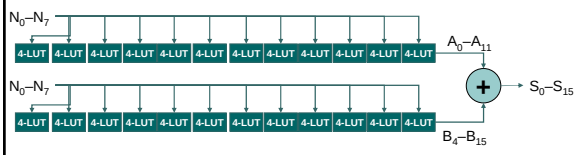
Arithmetic (cont.)

- Carry save multiplication



LUT-Based Constant Multipliers

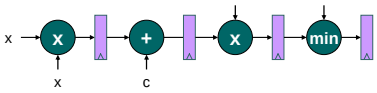
```
    10101011
  x NNNNNNNN
  ----------
    AAAAAAAAAAAA  (N * 1011 (LSN))
+ BBBBBBBBBBBB    (N * 1010 (MSN))
  ----------
  SSSSSSSSSSSSSSS  Product
```



- Constants can be changed in the LUTs to program new multipliers

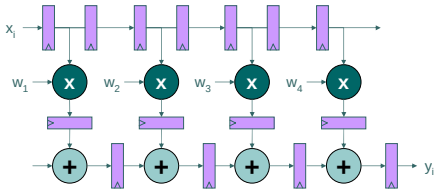
Systolic Architectures

- Goal – general methodology for mapping computations into hardware (spatial computing) structures
- Composition:
  - Simple compute cells (e.g. add, sub, max, min)
  - Regular interconnect pattern
  - Pipelined communication between cells
  - I/O at boundaries

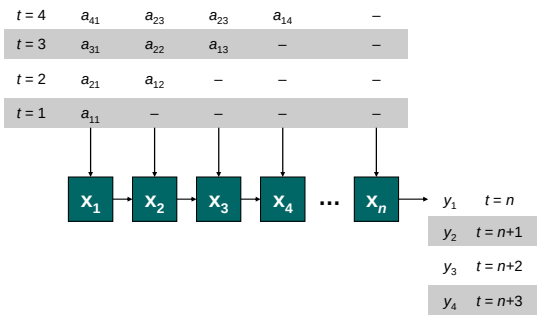


Finite Impulse Response

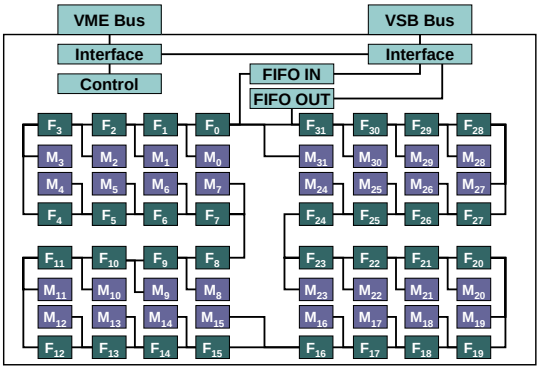
- Sequential
  - Memory bandwidth per output – $2k+1$
  - $O(k)$ cycles per output
  - $O(1)$ hardware
- Systolic
  - Memory bandwidth per output – 2
  - $O(1)$ cycles per output
  - $O(k)$ hardware



Matrix-Vector Product

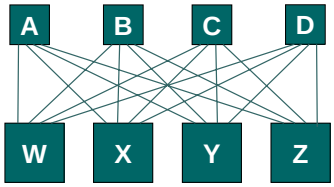


Splash 1 Architecture

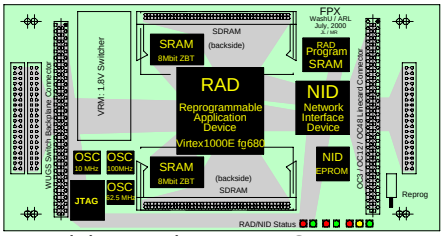


Other Multi-FPGA Topologies

- Crossbar topology:
  - Devices A-D are routing only
  - Gives predictable performance
  - Potential waste of resources for near-neighbor connections

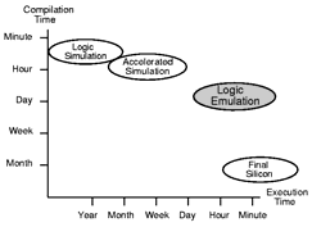


FPGA-based Router



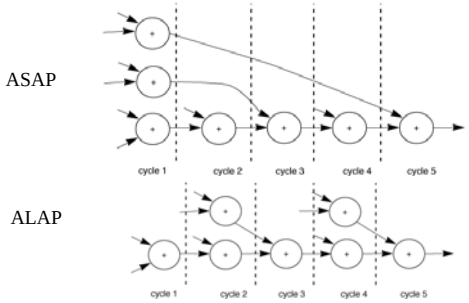
- FPX module contains two FPGAs
- NID – network interface device
  - Performs data queuing
- RAD – reprogrammable application device
  - Specialized control sequences

Logic Emulation

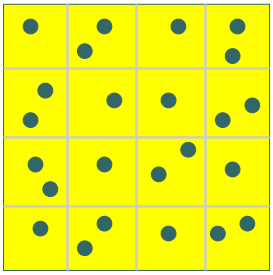


- Emulation takes a sizable amount of resources
- Compilation time can be large due to FPGA compiles

ASAP and ALAP Schedules



Recursive Partitioning



Next Steps

- VHDL for synthesis
- Non-conventional reconfigurable architectures
- HW/SW codesign / high-level compilation

- Other topics?
  - Second course survey next week
  - Provide general feedback, suggest additional topics

Midterm Exam

- Three questions
  - Review
  - Analysis
  - Extension
- Any paper mentioned in class is fair game
- Due in 1 week (10/16 – 12:00pm)
  - No class on Thursday!
- Some restrictions:
  - Work alone
  - Can ask if something is unclear ("what does this mean?" questions, not "how do I do this?" questions)
  - No late submissions – strict WebCT deadline