

ESE532: System-on-a-Chip Architecture

Day 8: September 29, 2021
Spatial Computations



Penn ESE532 Fall 2021 -- DeHon

Today

- Accelerator Pipelines (Part 1)
- FPGAs (Part 2)
- Computational Capacity (Part 3)
 - Zynq

Penn ESE532 Fall 2021 -- DeHon

2

Message

- Custom accelerators efficient for large computations
 - Exploit Instruction-level parallelism
 - Run many low-level operations in parallel
- Field-Programmable Gate Arrays (FPGAs)
 - Allow post-fabrication configuration of custom accelerator pipelines
 - Can offer high computational capacity

Penn ESE532 Fall 2021 -- DeHon

3

Accelerator Datapaths

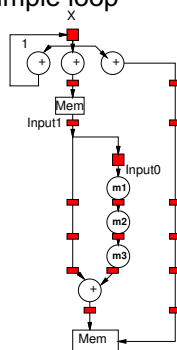
Penn ESE532 Fall 2021 -- DeHon

4

Pipeline Graph

- Last time: pipelined simple loop

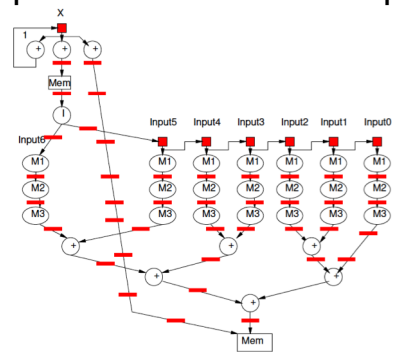
```
unsigned char Input0, Input1;
Input1=Input[Y * INPUT_WIDTH + X + 0];
for (int X = 0; X < OUTPUT_WIDTH; X++)
{
    unsigned int Sum;
    Input0=Input1;
    Input1=Input[Y * INPUT_WIDTH + X + 1];
    Sum = Coefficients_0 * Input0 + Input1;
    Output[Y * OUTPUT_WIDTH + X] = Sum;
}
```



Penn ESE532 Fall 2021 -- DeHon

5

Pipeline for Unrolled Loop



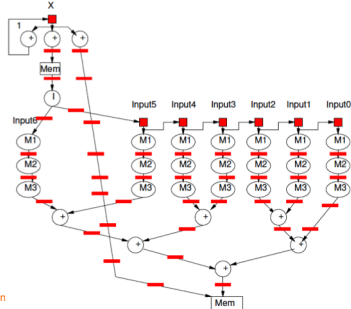
Penn ESE532 Fall 2021 -- DeHon

6

Preclass 1

- For fully unrolled loop shown, how many instructions per pipeline cycle?

- Add
- Mpy
- Load
- Store



Penn ESE532 Fall 2021 -- DeHon

7

Spatial Pipeline

- Can compute equivalent of tens of “instructions” in a cycle
- Wire up primitive operators
 - No indirection through register file, memory
- Pipeline for operator latencies
- Any dataflow graph of computational operations

Penn ESE532 Fall 2021 -- DeHon

8

Operators

- Can assemble any custom operators
 - Ones may not have in generic processor
- Processor
 - Add, bitwise-xor/and/or
 - Maybe: floating-point add, multiply
- Less likely
 - Square-root, exponent, cosine, encryption (AES) step, polynomial evaluate, log-number-system

Penn ESE532 Fall 2021 -- DeHon

9

Accelerators

- Compression/decompression
- Encryption/decryption
- Encoding (ECC, Checksum)
- Discrete Cosine Transform (DCT)
- Sorter
- Taylor Series Approximation of function
- Transistor evaluator
- Tensor or Neural Network evaluator

Penn ESE532 Fall 2021 -- DeHon

10

Streaming Dataflow

- Replace operator with custom accelerator
- Stream data to/from it

Penn ESE532 Fall 2021 -- DeHon

11

Streaming Dataflow Example



Penn ESE532 Fall 2021 -- DeHon

12

Application-Specific SoCs

- For dedicated applications may build custom hardware for accelerators
 - Layout VLSI, fab unique chips
 - ESE370, 570
- Video-encoder – include custom DCT, motion-estimation engines

Penn ESE532 Fall 2021 – DeHon

13

Apple A14 Bionic

- 88mm², 5nm
- 11.8 Billion Tr.
- iPhone 12
- 6 ARM cores
 - 2 fast (2.9–3GHz)
 - 4 low energy
- 4 custom GPUs
- 16 Neural Engines
 - 11 Trillion ops/s?



Image from <https://www.extremetech.com/computing/318715-comparison-of-apple-m1-a14-shows-differences-in-soc-design-details>; <https://www.tomshardware.com/news/apple-a14-bionic-revealed>
<https://www.anandtech.com/show/16226/apple-silicon-m1-a14-deep-dive/2>

Penn ESE532 Fall 2021 – DeHon

14

Customizable Accelerators

- With post-fabrication configurability can exploit without unique fabrication
- Need programmable substrate that allows us to wire-up computations

Penn ESE532 Fall 2021 – DeHon

15

Field-Programmable Gate Arrays

FPGAs
Part 2

Penn ESE532 Fall 2021 – DeHon

16

FPGA

- Idea: Can wire up programmable gates in the “field”
 - After fabrication
 - At your desk
 - When part “boots”
- Like a “Gate Array”
 - But not hardwired

Penn ESE532 Fall 2021 – DeHon

17

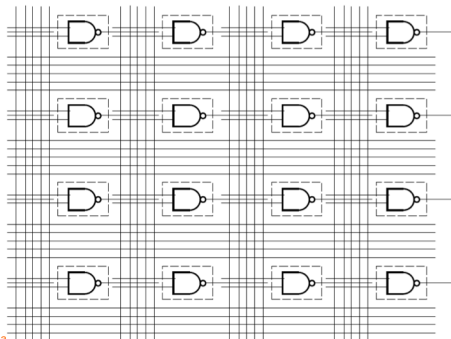
Gate Array

- Idea: Provide a collection of uncommitted gates
- Create your “custom” logic by wiring together the gates
- Less layout, fewer masks than full custom
 - Since only wiring together pre-fab gates
 - lower cost (fewer masks)
 - lower manufacturing delay

Penn ESE532 Fall 2021 – DeHon

18

Gate Array



Penn ESE532 Fa

9

GA → FPGA

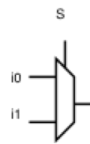
- Remove the need to even fabricate the wiring mask
- Make “customization” soft
- Key trick:
 - Use reprogrammable configuration bits
 - Typically: static-RAM bits
 - Like SRAM cells or latches in memory
 - Hold a configuration value

Penn ESE532 Fall 2021 – DeHon

20

Multiplexer Gate

- MUX
 - When $S=0$, output= i_0
 - When $S=1$, output= i_1

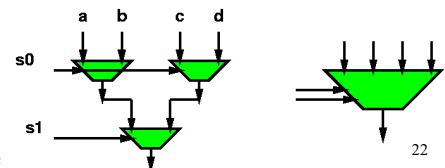


$$\text{Out} = \neg s \cdot i_0 + s \cdot i_1$$

Penn ESE532 Fall 2021 – DeHon

21

Mux with configuration bits = programmable gate

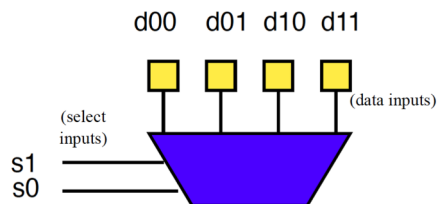


Penn ESE532 Fall 202

22

Preclass 4a

- How do we program to behave as and2?

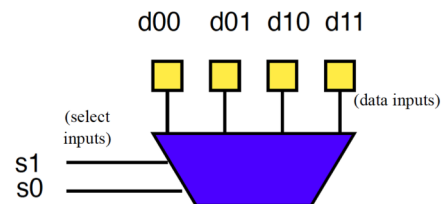


Penn ESE532 Fall 2021 – DeHon

23

Preclass 4b

- How do we program to behave as xor2?

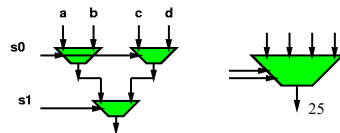


Penn ESE532 Fall 2021 – DeHon

24

Mux as Logic

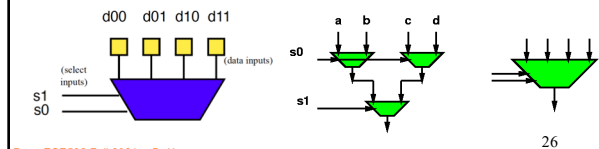
- Just by “configuring” data into this mux4,
 - Can select **any** two input function



Penn ESE532 Fall 2021 -- DeHon

LUT – LookUp Table

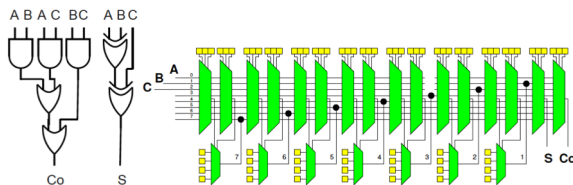
- When use a mux as programmable gate
 - Call it a **LookUp Table (LUT)**
 - Implementing the Truth Table for small # of inputs
 - # of inputs = k (need $\text{mux-}2^k$)
 - Just lookup the output result in the table



Penn ESE532 Fall 2021 -- DeHon

Preclass 6

- How do we program full adder?



Penn ESE532 Fall 2021 -- DeHon

27

FPGA

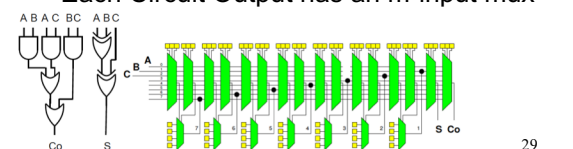
- Programmable gates + wiring
 - (both built from muxes w/ config. bits)
- Can wire up any collection of gates
 - Like a gate array

Penn ESE532 Fall 2021 -- DeHon

28

Simplistic FPGA (illustrate possibility)

- Every LUT input has a mux
- Every such mux has $m=(N+1)$ inputs
 - An input for each LUT output (N 2-LUTs)
 - An input for each Circuit Input (I Circuit inputs)
- Each Circuit Output has an m -input mux

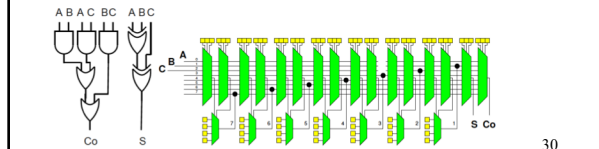


Penn

29

Simplistic FPGA (illustrate possibility)

- N 2-LUTs, I Circuit Inputs, O Circuit Outputs
- $2N+O$ muxes to connect
- Can build **any** combinational logic circuit that doesn't need more than N 2-input gates, I inputs, O outputs



Penn ESE532 Fall 2021 -- DeHon

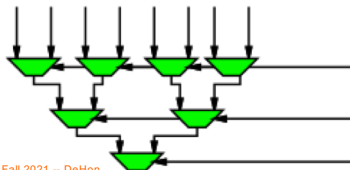
30

Preclass 3

How big is an m-input mux?

- In terms of 2-input muxes?
 - Warmup: how many for 4-input (Preclass 2)
 - Warmup: how many for 8-input (below)

m inputs – what we are selecting from



$\log_2(m)$ bits
to select
which input
routed to
output

Penn ESE532 Fall 2021 – DeHon

31

Math: Series Sums

- $A_0(1+r+r^2+r^3+r^4+\dots)$
- $A_0(1+r+r^2+r^3+r^4+\dots)*(1-r)$

$$= A_0 + A_0 r + A_0 r^2 + A_0 r^3 + A_0 r^4 + \dots$$

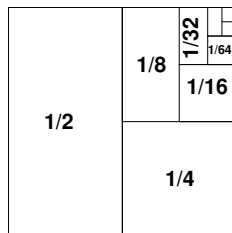
$$- A_0 r - A_0 r^2 - A_0 r^3 - A_0 r^4 - \dots$$

$$= A_0 \quad (\text{when } r < 1)$$
- $A_0(1+r+r^2+r^3+r^4+\dots)*(1-r) = A_0$
- $A_0(1+r+r^2+r^3+r^4+\dots) = A_0/(1-r)$

Penn ESE532 Fall 2021 – DeHon

32

Receding Sum



Penn ESE532 Fall 2021 – DeHon

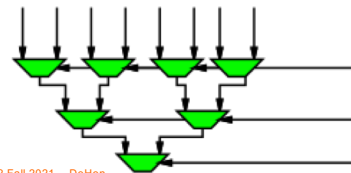
33

Preclass 3

How big is an m-input mux?

- M-input mux will require m-1 2-input muxes

m inputs – what we are selecting from



$\log_2(m)$ bits
to select
which input
routed to
output

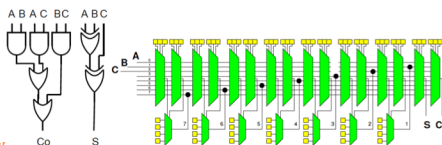
Penn ESE532 Fall 2021 – DeHon

34

Simplistic FPGA

(illustrate possibility...and expense)

- $2N+O$ m-input muxes; $m=N+1$
- Each m-input mux is m-1 2-input muxes
- Requires: $(2N+O)*(N+1-1)$ 2-input muxes
- Mux area grows as $\sim N^2$
 - when gate (LUT) area grows as N



Penn ESE

35

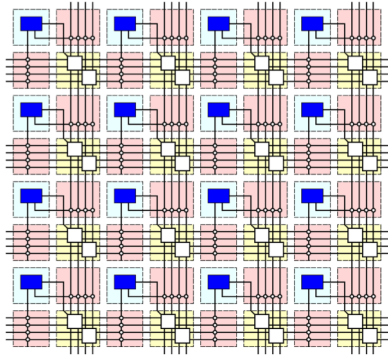
Interconnect

- Fully connected mux input is too expensive, growing as N^2
 - ...and not necessary
- Want
 - To be able to wire up gates
 - Economical with wires and muxes
 - ...and configuration bits
 - Exploit locality (keep wires short)

Penn ESE532 Fall 2021 – DeHon

36

Simple FPGA

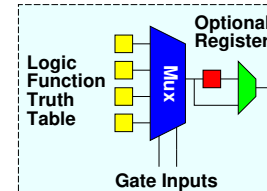


Penn ESE532 Fall 2021 -- DeHon

37

Register

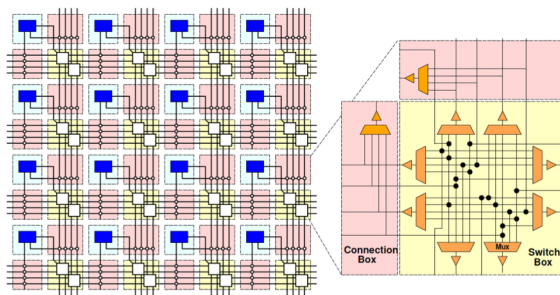
- Want to be able to pipeline logic
- ...and generally hold state
 - E.g. implement hold Input-N in preclass 1
- Add optional register on each gate



Penn ESE532 Fall 2021 -- DeHon

38

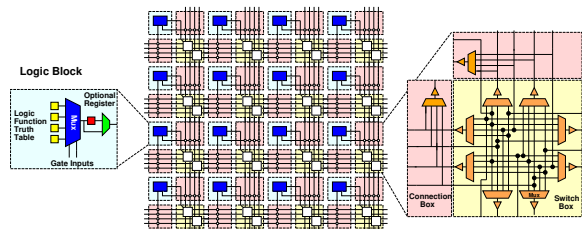
Simple FPGA



Penn ESE532 Fall 2021 -- DeHon

39

Simple FPGA



Penn ESE532 Fall 2021 -- DeHon

40

FPGA Design

- Raises many architectural design questions
 - How big (many inputs) should the gates have?
 - Are LUTs really the right thing...
 - How rich is the interconnect?
 - Wires/channel
 - Wire length
 - Switching options

Penn ESE532 Fall 2021 -- DeHon

41

Modern FPGAs

- Logic Blocks
 - hardwired fast-carry logic
 - Can implement adder bit in single "LUT"
 - Speed optimized: 6-LUTs
 - Energy, Cost optimization: 4-LUTs
 - Clusters many LUTs into a tile
- Interconnect
 - Mesh, segments of length 4 and longer

Penn ESE532 Fall 2021 -- DeHon

42

More than LUTs

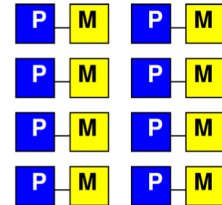
- Should there be more than LUTs in the “array” fabric?
- What else might we want?

Penn ESE532 Fall 2021 -- DeHon

43

Embedded Memory

- One flip-flop per LUT doesn't store state densely
- Want memory close to logic



Penn ESE532 Fall 2021 -- DeHon

44

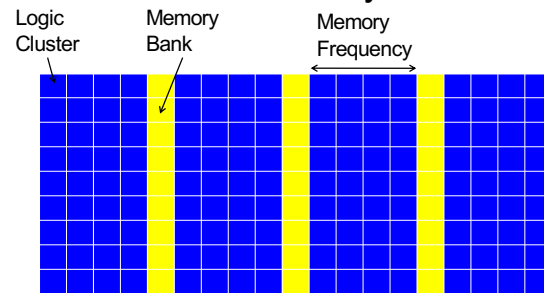
Embed Memory in Array

- Replace logic clusters
- Convenient to replace columns
 - Since area of memory may not match area of logic cluster

Penn ESE532 Fall 2021 -- DeHon

45

Embedded Memory in FPGA



Memory banks on Xilinx called BRAMs (Block RAMs)

Penn ESE532 Fall 2021 -- DeHon

46

Hardwired Multipliers

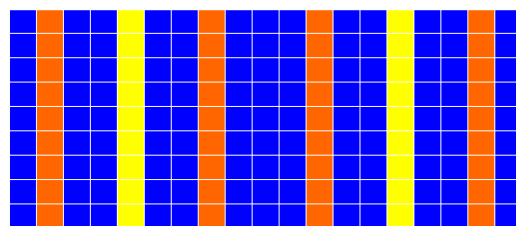
- Can build multipliers out of LUTs
 - Just as can implement multiplies on processor out of adds
- But, custom multiplier is smaller than LUT-configured multiplier
 - ...and multipliers common in signal processing, scientific/engineering compute

Penn ESE532 Fall 2021 -- DeHon

47

Multiplier Integration

- Integrate like memories
 - Replace columns



Penn ESE532 Fall 2021 -- DeHon

More FPGA Architecture Design Questions

- Size of Memories? Multipliers?
- Mix of LUTs, Memories, Multipliers?
- Add processors? Floating-point?
- Other hardwired blocks?
- How manage configuration?

Penn ESE532 Fall 2021 -- DeHon

49

Midterm (10/6 – next Wed.)

- Analysis
 - Bottleneck
 - Amdahl's Law
 - Speedup
 - Computational requirements
 - Resource Bounds
 - Critical Path
 - Latency/throughput/II
- Will be calculating/estimating runtimes
- From Code
- Forms of Parallelism
- Dataflow, SIMD, hardware pipeline, threads
- Pipelining/Retiming
- Map/schedule task graph to (multiple) target substrates
- Memory assignment and movement
- Area-time points 50

Penn ESE532 Fall 2021 -- DeHon

Midterm

- In-class exam
- Closed book, notes, etc.
- Calculators allowed (encouraged)
- Read midterm details posted on web
- Last five midterms, finals online
 - Both without answers (for practice)
 - ...and with answers (check yourself)
 - Check syllabus for previous terms
 - (but note last year online, open-book)
- Midterm comes earlier last year and this₁

Penn ESE532 Fall 2021 -- DeHon

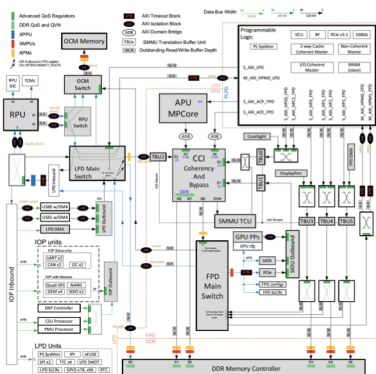
Zynq MPSoC

Part 3

Penn ESE532 Fall 2021 -- DeHon

52

Programmable SoC



UG1085
Xilinx
UltraScale
Zynq
TRM
(p27)

53

Penn ESE532 I

ZU3EG (Ultra96)

- 6-LUTs: 70,560
- DSP Blocks: 360
 - 18x27 multiply, 48b accumulate
- Block RAMs (BRAMs): 216
 - 36Kb
 - Dual port
 - Up to 72b wide (512x72)

Penn ESE532 Fall 2021 -- DeHon

54

DSP48

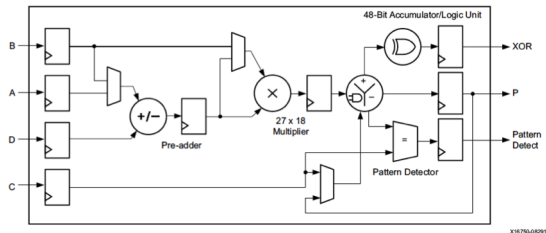


Figure 1-1: Basic DSP48E2 Functionality

Xilinx UG579 UltraScale DSP Slice User's Guide 55

Penn ESE532 Fall 2021 -- DeHon

Preclass 5

Approximating	Resources	Cycle	Per second
Zynq LUTs	70,000 adder bits	0.5 GHz	
4x ARM Scalar	4x2x64 adder bits	1.2 GHz	
4x ARM Neon	4x1x64 adder bits	1.2 GHz	
Zynq DSP	360 multiply-accumulates	0.5 GHz	
4x ARM Scalar	4x(1 mpy+1add)	1.2 GHz	
4x ARM Neon	4x1x4 multiply-accumulates	1.2 GHz	

- How compare between ARM scalar, ARM NEON and FPGA array?

– Adder-bits/second?

– Multiply-accumulators/second?

56

Penn ESE532 Fall 2021 -- DeHon

Capacity → Density

- Says Zynq has high computational capacity in FPGA
- More broadly
 - FPGA can have more compute/area than processor
 - E.g., more adder bits in some fixed area
 - SIMD can have more compute/area than processor (Day 6)
 - How wide SIMD can you exploit?

57

Penn ESE532 Fall 2021 -- DeHon

FPGA Potential

- FPGA Array has high raw capacity
- Exploitable when computation has high regularity
 - Uses the same computation over-and-over
 - High throughput on a computation
 - Build customized accelerator pipeline to match the computation
- Low-hanging fruit
 - Operator/function takes most of the compute time

58

Penn ESE532 Fall 2021 -- DeHon

90/10 Rule

- Observation that code is not used uniformly
- 90% of the time is spent in 10% of the code
- Knuth: 50% of the time in 2% of the code
- Opportunity
 - Build custom datapath in FPGA (hardware) for that 10% (or 2%) of the code

59

Penn ESE532 Fall 2021 -- DeHon

Big Ideas

- Custom accelerators efficient for large computations
 - Exploit Instruction-level parallelism
 - Run many low-level operations in parallel
- Field Programmable Gate Arrays (FPGAs)
 - Allow post-fabrication configuration of custom accelerator pipelines
 - Can offer high computational capacity

60

Penn ESE532 Fall 2021 -- DeHon

Admin

- Reading for Day 9 on canvas
- HW4 due on Friday
- Midterm on Wednesday
 - No assignment due on Friday (10/8)
 - Previous midterms (with solutions) on web syllabus of previous years
- HW5 out
 - Heavier – start early...have more than week
 - Vivado HLS synthesis slow (plan for it)
 - Due Wed. 10/13 (before Fall Break)