

# ESE5320: System-on-a-Chip Architecture

Day 11: October 7, 2024  
Data Movement  
(Interconnect, DMA)

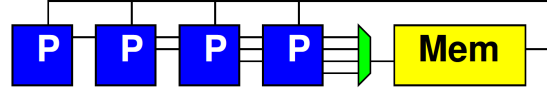


Penn ESE5320 Fall 2024 -- DeHon

1

## Preclass 1

- N processors
- Each: 1 read, 10 cycle compute, 1 write
- Memory: 1 read or write per cycle
- How many processors can support before saturate memory capacity?



Penn ESE5320 Fall 2024 -- DeHon

2

2

## Schedule Memory Port

12	13	14	15	16	17	18	19	20	21	22	23	24	25
P1.1 write	P1.2 read	P2.1 write	P2.2 read	P3.1 write	P3.2 read	P4.1 write	P4.2 read	P5.1 write	P5.2 read	P6.1 write	P6.2 read	P1.2 write	P1.3 read
P1 compute f on 2 <sup>nd</sup> iteration													
P2 compute f on 2 <sup>nd</sup> iteration													

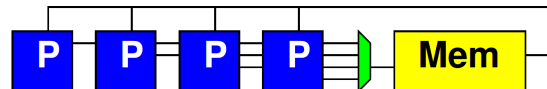
Penn ESE5320 Fall 2024 -- DeHon

3

3

## Bottleneck

- Sequential access to a common memory can become the bottleneck



Penn ESE5320 Fall 2024 -- DeHon

4

4

## Previously

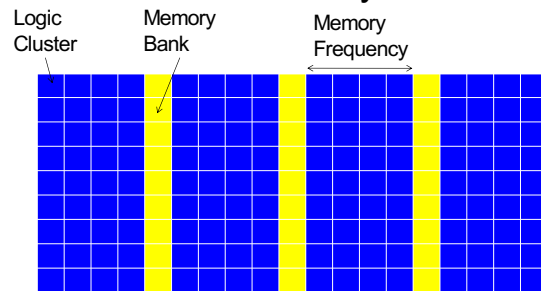
- Want data in small memories
  - Low latency, high bandwidth
- FPGA has many memories all over fabric

Penn ESE5320 Fall 2024 -- DeHon

5

5

## Embedded Memory in FPGA



ZU3EG (Ultra96) has 216 36Kb BRAMs  
VU9P (Amazon F1) has 2,160

Penn ESE5320 Fall 2024 -- DeHon

6

6

## Previously

- Want data in small memories
  - Low latency, high bandwidth
- FPGA has many memories all over fabric
- Want C arrays in small memories
  - Partitioned so can perform enough reads (writes) in a cycle to avoid memory bottleneck

Penn ESE5320 Fall 2024 -- DeHon

7

7

## Today

- Interconnect Infrastructure (Part 1)
- Peripherals (Part 2)
- Data Movement Threads (Part 3)
- DMA -- Direct Memory Access (Part 4)

Penn ESE5320 Fall 2024 -- DeHon

8

8

## Message

- Need to move data
- Often use shared interconnect to make physical connections
- Useful to move data as separate thread of control
  - Dedicated a processor is inefficient
  - Useful to have dedicated data-movement hardware: Direct Memory Access (DMA)

Penn ESE5320 Fall 2024 -- DeHon

9

9

## Term: Peripheral

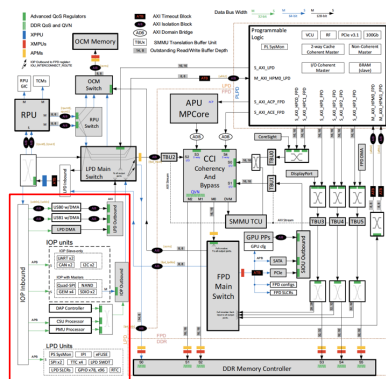
- “On the edge (or periphery) of something”
- Peripheral device – device used to put information onto or get information off of a computer
  - E.g.
    - Keyboard, mouse, modem, USB flash drive, ...

Penn ESE5320 Fall 2024 -- DeHon

10

10

## Programmable SoC



UG1085  
Xilinx  
UltraScale  
Zynq  
TRM  
(p27)

Penn ESE5320

11

11

## Memory and I/O Organization

- Architecture contains
  - Large memories
    - For density, necessary sharing
  - Small memories local to compute
    - For high bandwidth, low latency, low energy
  - **Peripherals** for I/O
- Need to move data
  - Among memories and I/O
    - Large to small and back
    - Among small
    - From Inputs, To Outputs

Penn ESE5320 Fall 2024 -- DeHon

12

12

## How move data?

- Abstractly, using stream links.
- Connect stream between producer and consumer.
- Ideally: dedicated wires

Penn ESE5320 Fall 2024 -- DeHon 13

13

## Dedicated Wires?

- What might prevent us from having dedicated wires between all communicating units?

Penn ESE5320 Fall 2024 -- DeHon 14

14

## Making Connections

- Cannot always be dedicated wires
  - Programmable
  - Wires take up area
  - Don't always have enough traffic to consume the bandwidth of point-to-point wire
  - May need to serialize use of resource
    - E.g. one memory read per cycle
  - Source or destination may be sequentalized on hardware

Penn ESE5320 Fall 2024 -- DeHon 15

15

## Model

- Programmable, possibly shared interconnect

Penn ESE5320 Fall 2024 -- DeHon 16

16

## Simple Realization

Shared Bus

- Write to bus with address of destination
- When address match, take value off bus
- Pros?
- Cons?

Penn ESE5320 Fall 2024 -- DeHon 17

17

## Alternate: Crossbar

- Provide programmable connection between all sources and destinations
- Any destination can be connected to any single source

Penn ESE5320 Fall 2024 -- DeHon 18

18

## Simplistic FPGA (illustrate possibility) Day 8

- 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 ESE5320 Fall 2024 -- DeHon

19

## Alternate: Crossbar

- Provide programmable connection between all sources and destinations
- Any destination can be connected to any single source

Penn ESE5320 Fall 2024 -- DeHon

20

## Crossbar

Penn ESE5320 Fall 2024 -- DeHon

21

## Preclass 2

- K-input, O-output Crossbar
- How many 2-input muxes?

Penn ESE5320 Fall 2024 -- DeHon

22

## Crossbar

- Provides high bandwidth
  - Minimal blocking
- Costs large amounts of area
  - Grows fast with inputs, outputs

Penn ESE5320 Fall 2024 -- DeHon

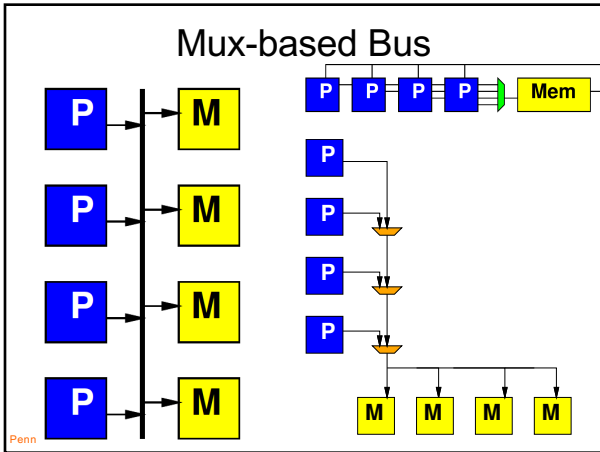
23

## General Interconnect

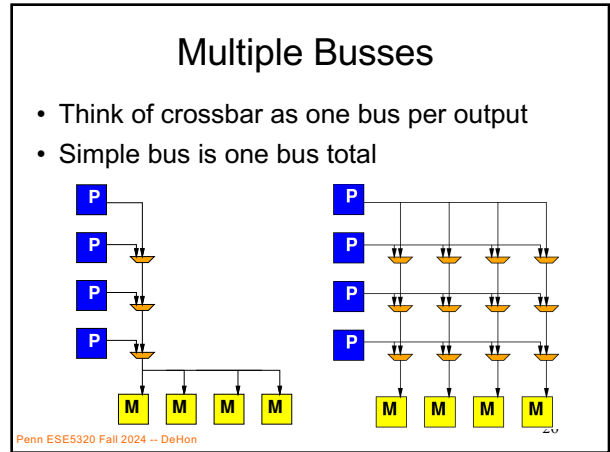
- Generally, want to be able to parameterize designs
- Here: tune area-bandwidth
  - Control how much bandwidth provide

Penn ESE5320 Fall 2024 -- DeHon

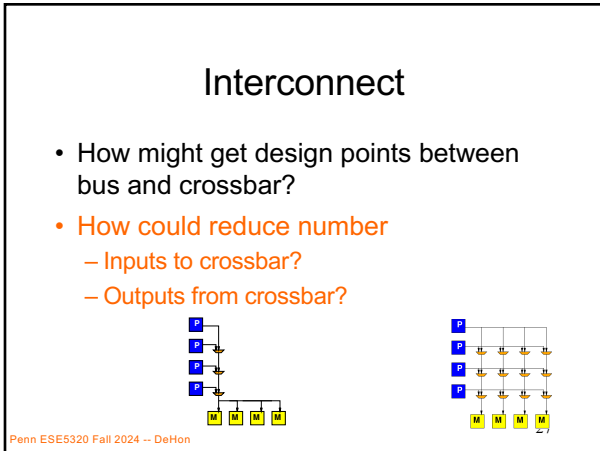
24



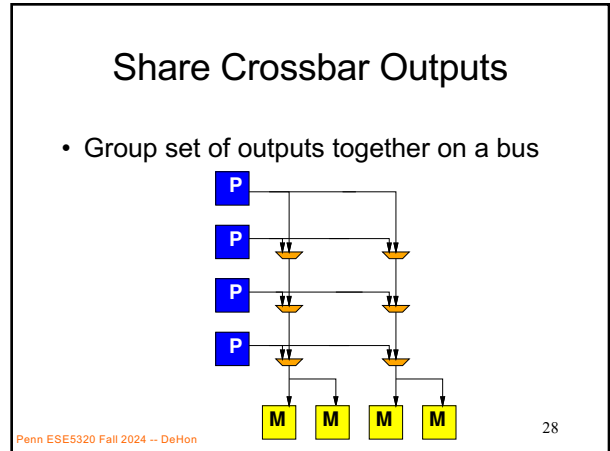
25



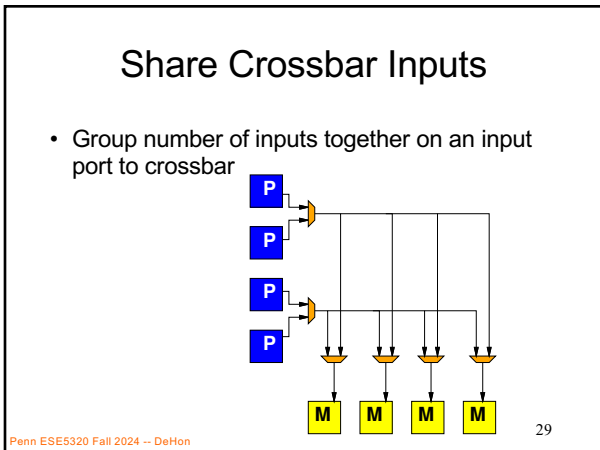
26



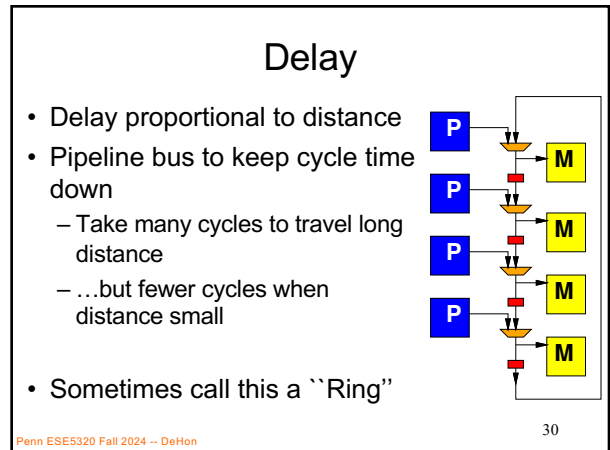
27



28



29



30

### Local Interconnect

- How many cycles from:
  - PE3 to PE2
  - PE3 to PE1
  - PE3 to PE4

Penn ESE5320 Fall 2024 -- DeHon

31

### Mesh

Mesh is a set of Horizontal and Vertical Rings with Option to switch H to V

Penn ESE5320 Fall 2024 -- DeHon

32

### Mesh

- Delay Proportional to distance in 2D

Penn ESE5320 Fall 2024 -- DeHon

33

### Hierarchical Busses

Penn ESE5320 Fall 2024 -- DeHon

34

### Interconnect

- Will need an infrastructure for programmable connections
- Rich design space to tune area-bandwidth-locality

Penn ESE5320 Fall 2024 -- DeHon

35

### Peripherals

Part 2

Penn ESE5320 Fall 2024 -- DeHon

36

## Input and Output

- Typical SoC has I/O with external world
  - Sensors
  - Actuators
  - Keyboard/mouse, display
  - Communications
- Also accessible from interconnect

usb ethernet

Penn ESE5320 Fall 2024 -- DeHon

37

## Programmable SoC

UG1085  
Xilinx  
UltraScale  
Zynq  
TRM  
(p27)

Penn ESE5320

38

## High Speed I/O

UG1085  
Xilinx  
UltraScale  
Zynq  
TRM  
(p27)

Penn ESE5320 Fall 2024 -- DeHon

39

## Masters and Slaves

- Two kinds of entities on interconnect
- Master – can initiate requests
  - E.g. **processor** that can perform a read or write
- Slaves – can only respond to requests
  - E.g. **memory** that can return the read data from a read request

Penn ESE5320 Fall 2024 -- DeHon

40

## Simple Peripheral Model

- Peripherals are slave devices
  - Masters can read input data
  - Masters can write output data
  - To move data, master (e.g. processor) initiates

Penn ESE5320 Fall 2024 -- DeHon

41

## Simple Peripheral Model

- Peripherals are slave devices
  - Masters can read input data
  - Masters can write output data
  - To move data, master (e.g. processor) initiates
- Demanding processor touch every data item has some negative consequences

Penn ESE5320 Fall 2024 -- DeHon

42

## Timing Demands

- Must read each input before overwritten
- Must write each output within real-time window
- Must guarantee processor scheduled to service each I/O at appropriate frequency
- How many cycles between 32b input words for 1Gb/s network and 32b, 1GHz processor?
  - Consider input data shifted into register 1b per ns
  - Must read out 32b register before overwritten

Penn ESE5320 Fall 2024 -- DeHon 43

43

## Refine Model

- Give each peripheral local FIFO
- Processor must still move data
- For same input data rate, how does this change requirements and impact?
  - How many cycles can processor run between servicing inputs?

Penn ESE5320 Fall 2024 -- DeHon 44

44

## Long Latency Memory Operations

### Part 3

Penn ESE5320 Fall 2024 -- DeHon 45

45

## Day 3

- Large memories are slow
  - Latency increases with memory size
- Distant memories are high latency
  - Multiple clock-cycles to cross chip
  - Off-chip memories even higher latency

Penn ESE5320 Fall 2024 -- DeHon 46

46

## Day 3, Preclass 2

- 10 cycle latency to memory
- If must wait for data return, latency can degrade throughput
- 10 cycle latency + 10 op + (assorted)
  - More than 20 cycles / result

```

for(i=0;i<MAX;i++) {
  in=a[i]; // memory read
  out=f(in); // 10 cycle compute
  b[i]=out;
}
  
```

Penn ESE5320 Fall 2024 -- DeHon 7

47

## Preclass 3

- Throughput using 3 threads on 3 processors: P1, P2, P3?
  - Throughput of each?
  - Throughput of collection?

```

P1: for(i=0;i<MAX;i++) Astream.write(a[i]);
P2: while(1) {Astream.read(aval); Bstream.write(f(aval));}
P3: for(i=0;i<MAX;i++) Bstream.read(b[i]);
  
```

Penn ESE5320 Fall 2024 -- DeHon 48

48



## Fetch (Write) Threads

- Potentially useful to move data in separate thread
  - (loop that compiler implements as a thread)
- Especially when
  - Long (potentially variable) latency to data source (memory)
- Useful to split request/response

Penn ESE5320 Fall 2024 -- DeHon

49

49

## DMA Part 4

### Direct Memory Access

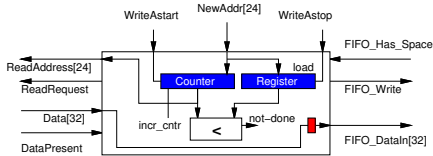
Penn ESE5320 Fall 2024 -- DeHon

50

50

## Preclass 4a

```
P1: for(i=0;i<MAX;i++) Astream.write(a[i]);
```



```
int *p;
P1: for(p=&(a[0]);p<&(a[MAX]);p++) Astream.write(*p);
```

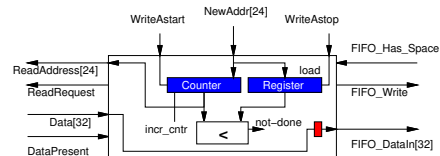
Penn ESE5320 Fall 2024 -- DeHon

51

51

## Preclass 4a

```
P1: for(i=0;i<MAX;i++) Astream.write(a[i]);
What NewAddr[24] provide for WriteAstart, Write Astop?
```



```
int *p;
P1: for(p=&(a[0]);p<&(a[MAX]);p++) Astream.write(*p);
```

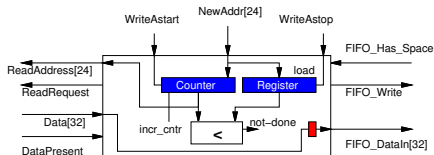
Penn ESE5320 Fall 2024 -- DeHon

52

52

## Preclass 4a

```
P1: for(i=0;i<MAX;i++) Astream.write(a[i]);
p<&(a[MAX]) p++
```



```
int *p;
P1: for(p=&(a[0]);p<&(a[MAX]);p++) Astream.write(*p);
```

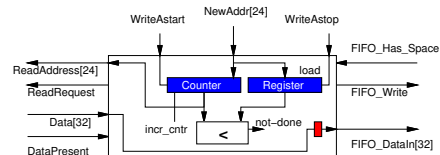
Penn ESE5320 Fall 2024 -- DeHon

53

53

## Preclass 4a

```
P1: for(i=0;i<MAX;i++) Astream.write(a[i]);
Read *p Astream.write
```



```
int *p;
P1: for(p=&(a[0]);p<&(a[MAX]);p++) Astream.write(*p);
```

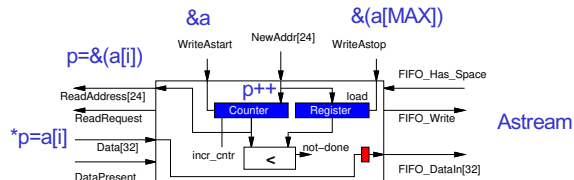
Penn ESE5320 Fall 2024 -- DeHon

54

54

## Preclass 4a

```
P1: for(i=0;i<MAX;i++) Astream.write(a[i]);
```



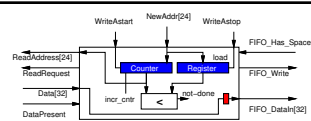
```
int *p;
P1: for(p=&(a[0]);p<&(a[MAX]);p++) Astream.write(*p);
```

Penn ESE5320 Fall 2024 -- DeHon

55

55

## Control Logic



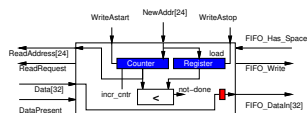
- FIFO\_Write =
- ReadRequest =
- Incr\_cnt =

Penn ESE5320 Fall 2024 -- DeHon

56

56

## Control Logic



- FIFO\_Write = DataPresent & FIFO\_Has\_Space
- ReadRequest = not-done
- Incr\_cnt = not-done
- (AckDataPresent = FIFO\_Write)

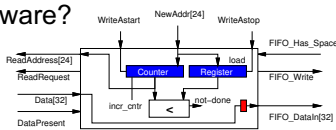
Penn ESE5320 Fall 2024 -- DeHon

57

57

## Preclass 4

- How much hardware?
  - Counter bits?
  - Registers?
  - Comparators?
  - Control Logic gates? (4cd)
- Compare to MicroBlaze
  - small RISC Processor optimized for Xilinx
  - minimum config 630 6-LUTs
  - Would take several cycles per word



Penn ESE5320 Fall 2024 -- DeHon

58

58

## Observe

- Modest hardware can serve as data movement thread
  - Much less hardware than a processor
  - Higher throughput than a processor
  - Offload work from processors
- Small hardware allow peripherals to be **master** devices on interconnect

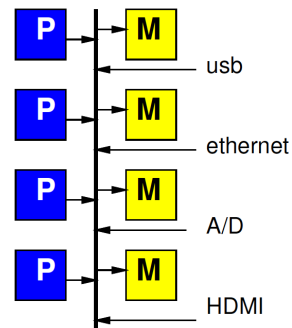
Penn ESE5320 Fall 2024 -- DeHon

59

59

## DMA

- Direct Memory Access (DMA)
- “Direct” – inputs (and outputs) don't have to be indirectly handled by the processor between memory and I/O
- I/O unit directly reads/writes memory



Penn ESE5320 Fall 2024 -- DeHon

60

60

## DMA

- Direct Memory Access (DMA)
- Peripheral as Master
  - Can write **directly** into (read from) memory
  - Saves processor from copying
  - Reduces demand to schedule processor to service

61

Penn ESE5320 Fall 2024 -- DeHon

61

## DMA Engine

- Data Movement Thread
  - Specialized Processor that moves data
  - Another heterogeneous processing engine
- Act independently (hence thread)
- Implement data movement
- Can build to move data between memories (Slave devices)
- E.g., Implement P1, P3 in Preclass 3

62

Penn ESE5320 Fall 2024 -- DeHon

62

## DMA Engine

63

Penn ESE5320 Fall 2024 -- DeHon

63

## Programmable DMA Engine

- What copy from?
- How much?
- Where copy to?
- Stride?
- What size data?
- Loop?
- Transfer Rate?

64

Penn ESE5320 Fall 2024 -- DeHon

64

## Multithreaded DMA Engine

- One copy task not necessarily saturate bandwidth of DMA Engine
- Can share engine performing many transfers (channels)
- Separate transfer state for each
  - Hence thread (or channel)
- Swap among threads
  - Simplest: round-robin:
    - 1, 2, 3, .. K, 1, 2, 3, .. K, 1, ..

65

Penn ESE5320 Fall 2024 -- DeHon

65

## Programmable SoC

UG1085  
Xilinx  
UltraScale  
Zynq  
TRM  
(p27)

66

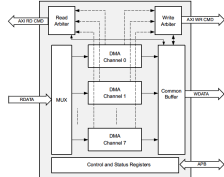
Penn ESE5320

66

## Hardwired and Programmable

- Zynq has hardwired DMA engine
  - 8 channels
- Also build data movement engines (Data Movers) in FPGA fabric

UG1085  
Xilinx  
UltraScale  
Zynq  
TRM  
Ch. 10  
(p519)



67

Penn ESE5320 Fall 2024 -- DeHon

67

## Example

- Networking Application: inline packet encrypt



- Header on processor
- Payload (encrypt, checksum) on FPGA
- DMA from ethernet → main memory
- DMA main memory → BRAM
- Stream between payload components
- DMA from encrypted packet and checksum to ethernet out

Penn ESE5320 Fall 2024 -- DeHon

68

## AXI

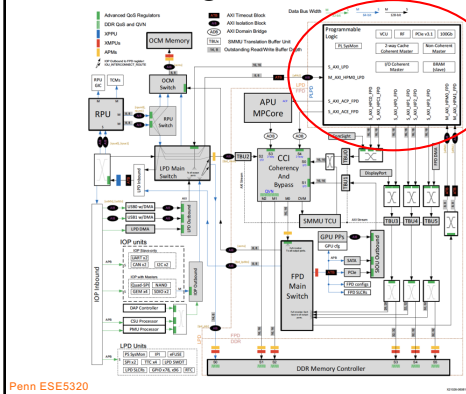
- Advanced eXtensible Interface
  - Originally developed by ARM
  - On-chip communication bus standard
  - Particular communication protocol
- Full AXI
  - Read/write operations with bursts
    - Burst = single address + length
    - Large, contiguous block of memory
  - Separate send/receive data channels
- AXI-S – for streaming connections
- AXI-lite – simpler, not burst

Penn ESE5320 Fall 2024 -- DeHon

69

69

## Programmable SoC



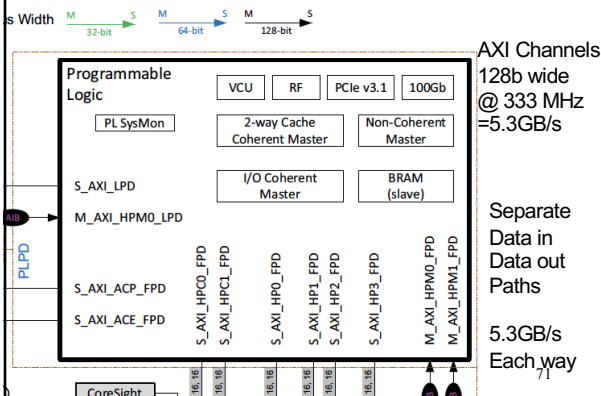
UG1085  
Xilinx  
UltraScale  
Zynq  
TRM  
(p27)

Penn ESE5320

70

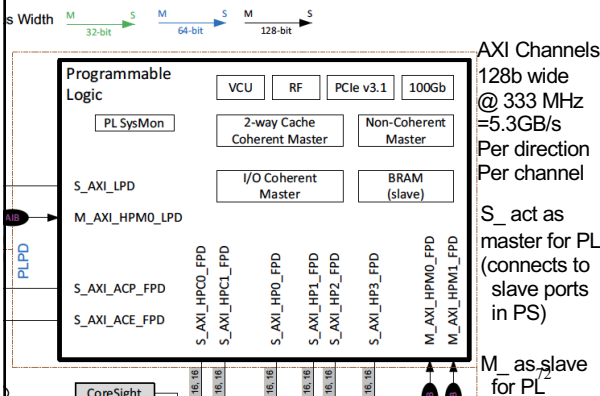
70

## Programmable SoC



71

## Programmable SoC



72

## DMA in Vitis

- Vitis/OpenCL demands that we write code to perform DMA of data to and from accelerators in FPGA fabric
- We will see specifics on Monday
- Have some options to control
  - With pragmas
  - With choice of data and burst sizes
  - Explore HW6

Penn ESE5320 Fall 2024 -- DeHon

73

73

## Big Ideas

- Need to move data
- Shared Interconnect to make physical connections – can tune area/bw/locality
- Useful to
  - move data as separate thread of control
  - Have dedicated data-movement hardware: DMA

Penn ESE5320 Fall 2024 -- DeHon

74

74

## Admin

- Feedback including lab
- Midterm Wednesday
  - Review Monday 8-9pm in Ketterer
  - Here classtime
  - Arrive early, start right at 10:15am
- HW5 due on Friday
- HW6 coming soon

Penn ESE5320 Fall 2024 -- DeHon

75

75