
ECE 636

Reconfigurable Computing

Lecture 6

Contrasting Processors: Fixed and Configurable



What is Computation?

- **Calculating predictable data outputs from data inputs.**

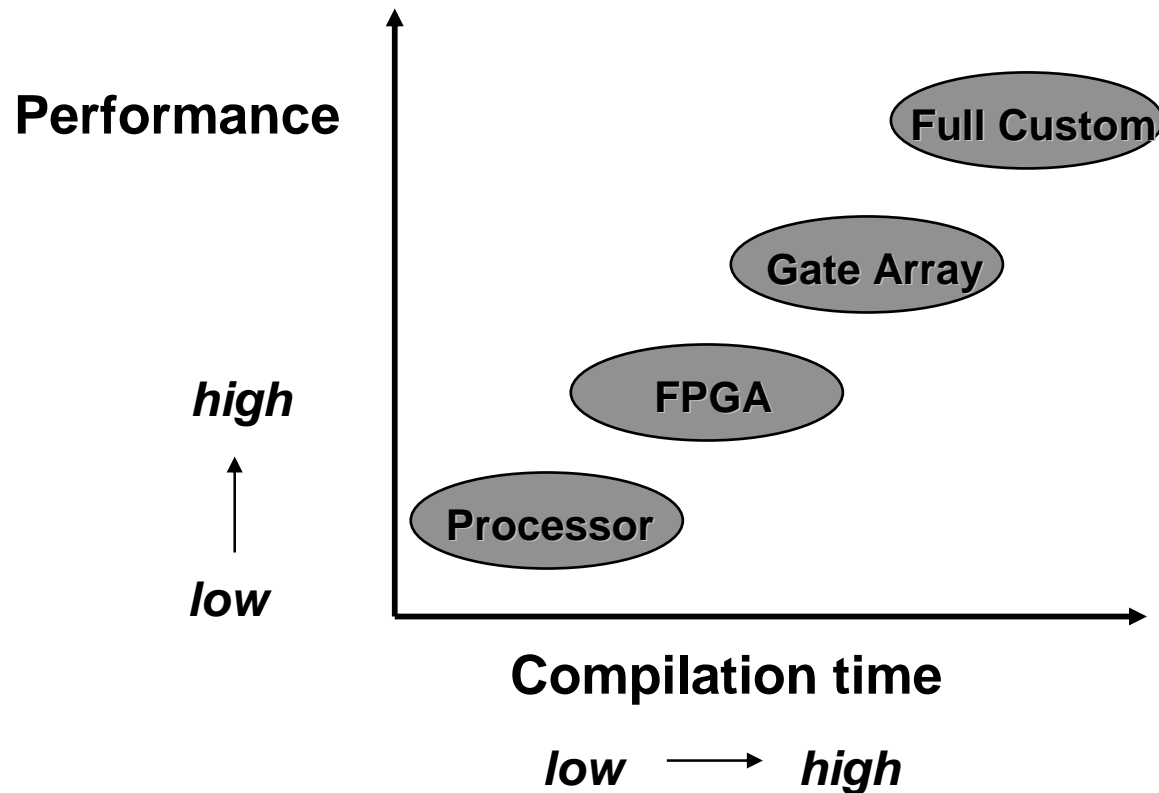
What should we expect from a computing device?

- **Gives correct answer.**
- **Takes up finite space**
- **Computes in finite time**
- **Can solve all problems?**
 - **Compilation**
 - **Implementation**

Other issues

Compilation

- How long does it take to “map” an idea to hardware?
- Why is the processor so “easy” to target for compilation?



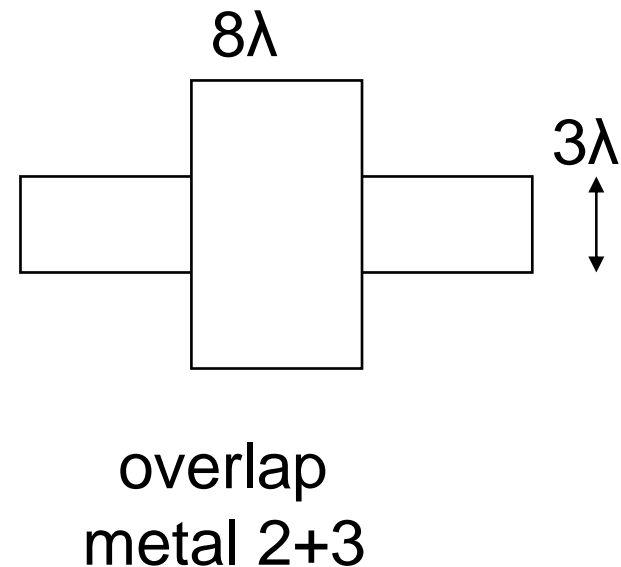
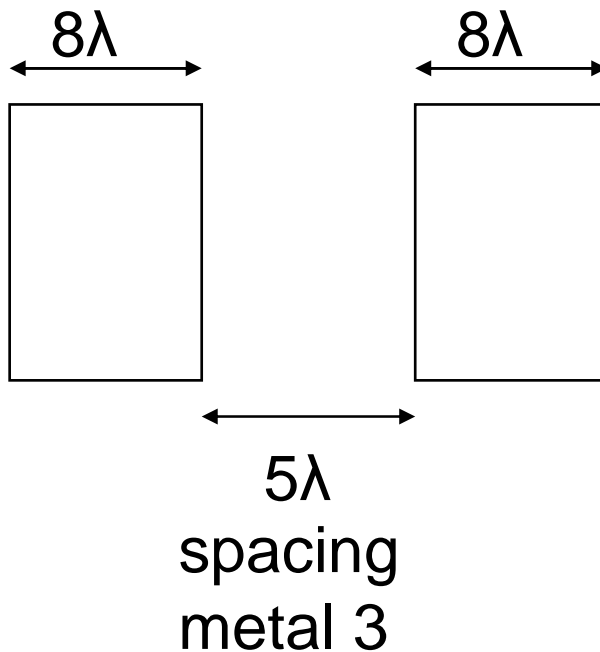
What are variables in Computation?

- *Time* -> How long does it take to compute the answer?
 - *Area* -> How much silicon space is required to determined the answer?
-
- Processor generally fixes computing area. Problem evaluated over time through instructions.

 - FPGA can create flexible amount of computing area. Effectively, the configuration memory is the computing instruction.

Measuring Feature Size

- Current FPGAs follow the same technology curve as microprocessors.
- Difficult to compare device sizes across generations so we use a fixed metric, lambda (λ).
- Lambda defines basic feature sizes in the VLSI device.



Toward Computational Comparison

Dehon metrics:

Computational density of a device

4 input gate-evaluations

$$\lambda^2 \times s$$

Processor:

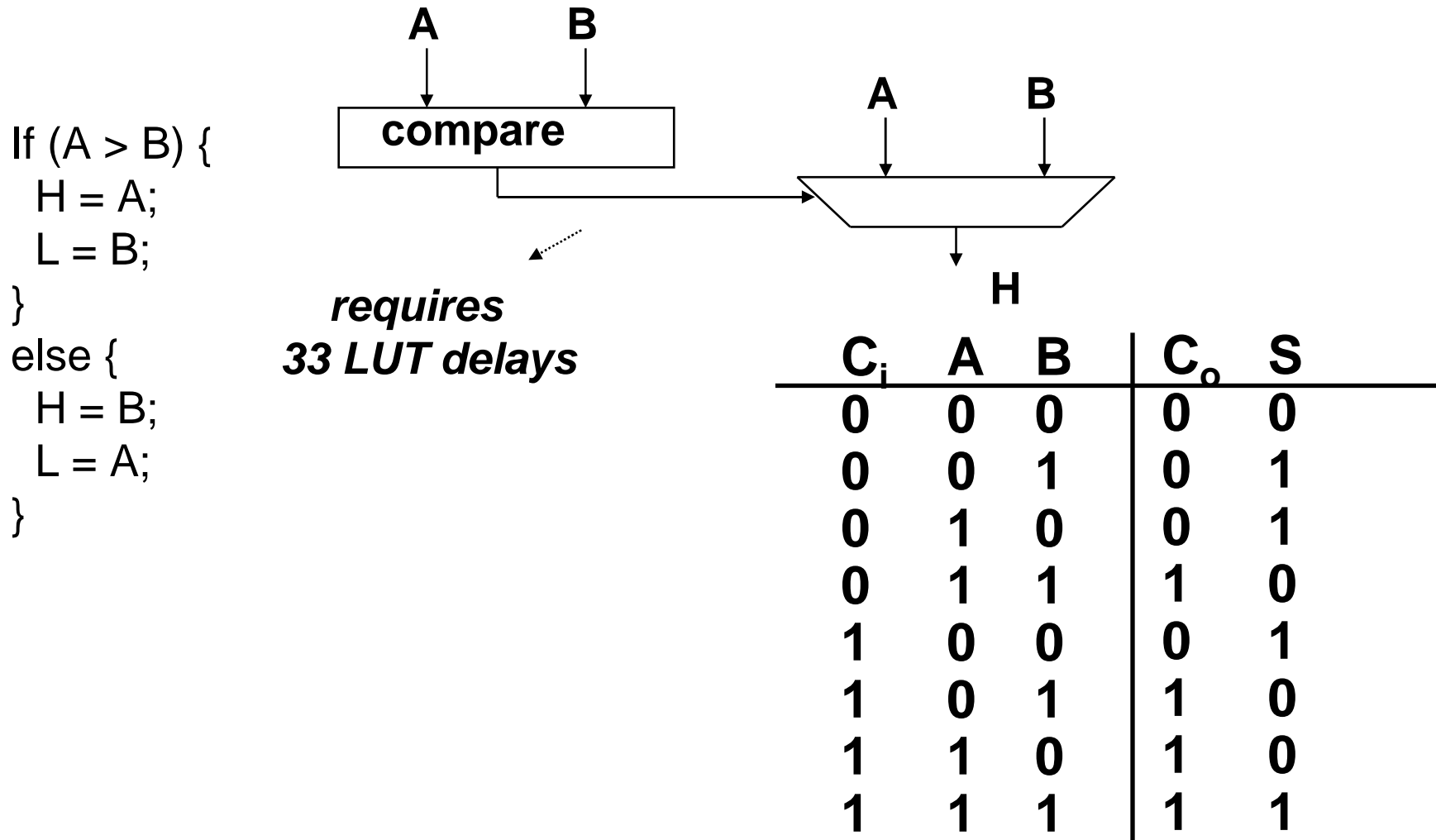
$$\frac{2 \times N_{ALU} \times W_{ALU}}{A_{proc} \times t_{cycle}}$$

FPGA:

$$\frac{N_{4lut}}{A_{array} \times t_{cycle}}$$

Degradation

- FPGA can't really be clocked at 1/7 ns due to interconnect.
- Consider the Bubblesort block from the first class.



New Comparison

Design	organization	λ^2	cycle	ge/ $\lambda^2 \times s$
1994 MIPS	1x32	1.7G	2 ns	19
1992 Xilinx	49 CLB (2 x4LUT)	61M	7 ns	230

- Processor required three cycles at 500 MHz
- FPGA requires 33 LUTs delays per computation.

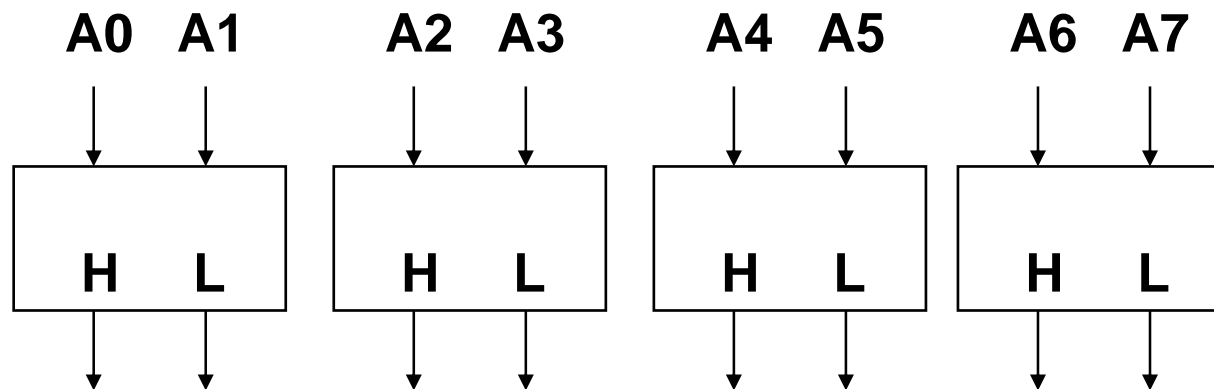
Could consider other parts of design.

Parallelization

- How this performance factor change over time? – through parallelization.

For a given operation $ge/(\lambda^2.s)$ seems the same -> 7

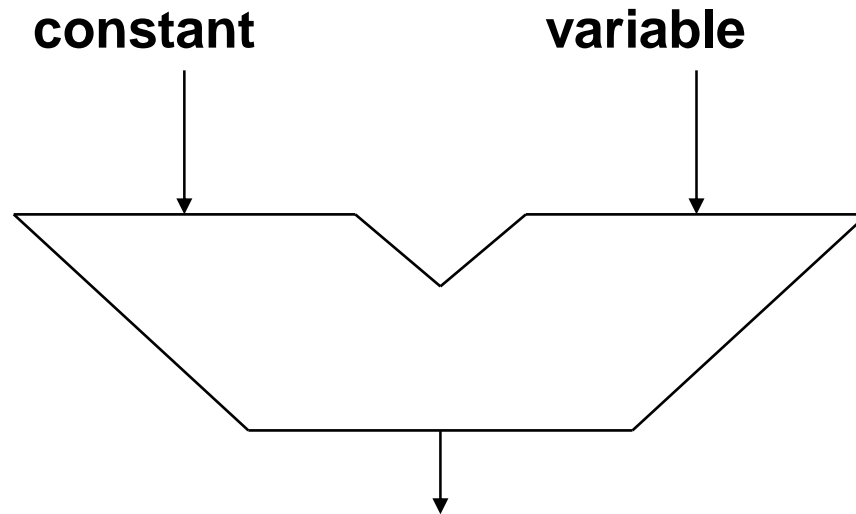
However, multiple comparisons could be performed in parallel.



Now FPGA metric is 28

Of course, device may be only partially filled.

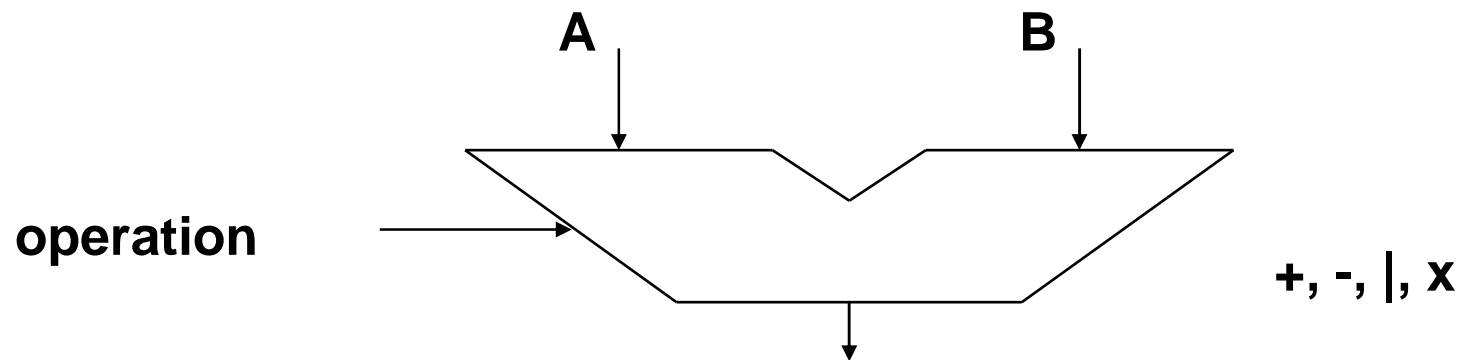
Specialization



- **Example: encryption**

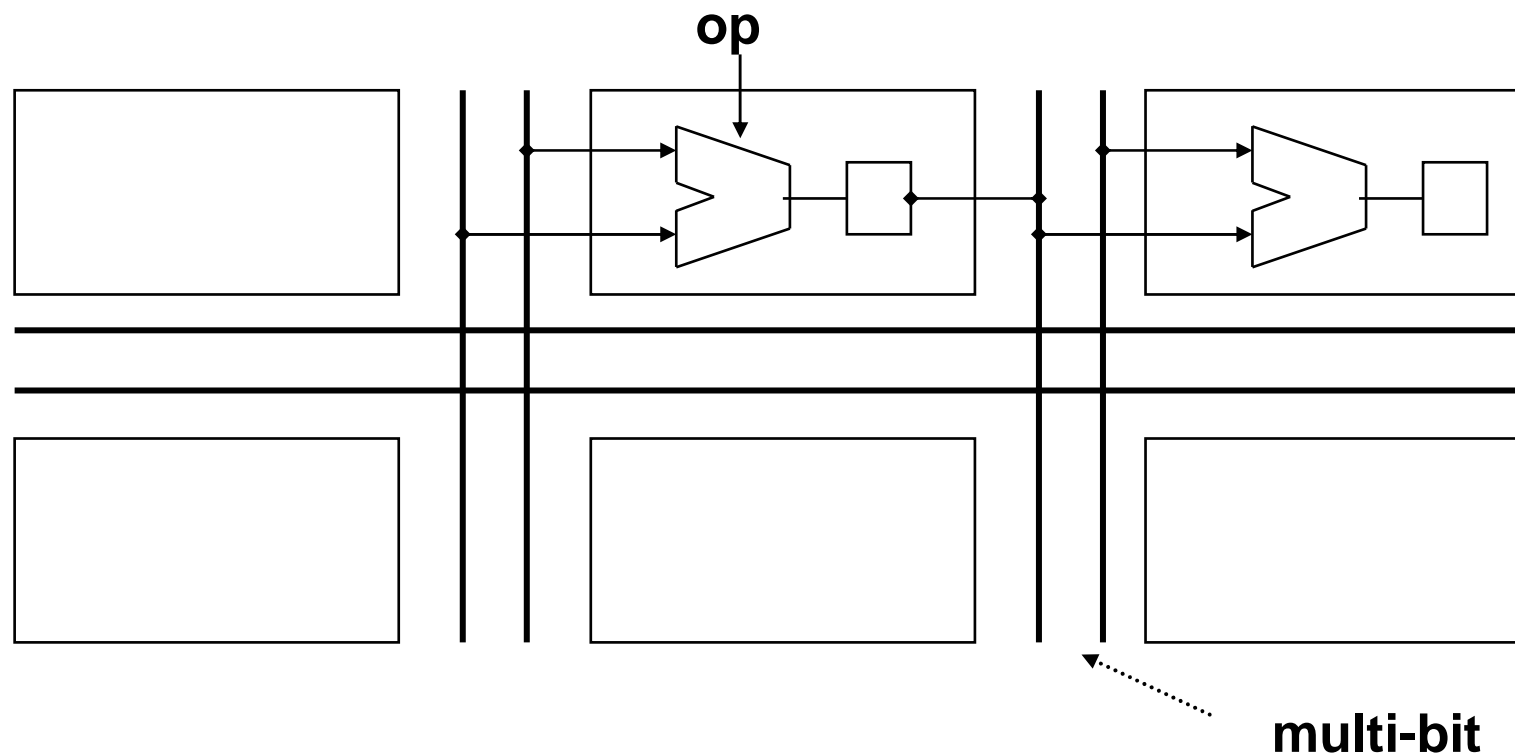
Instructions

- Many applications have little parallelism or have variable hardware requirements during execution.
- Here using more area doesn't increase computational density.
- Better to reuse hardware through instructions

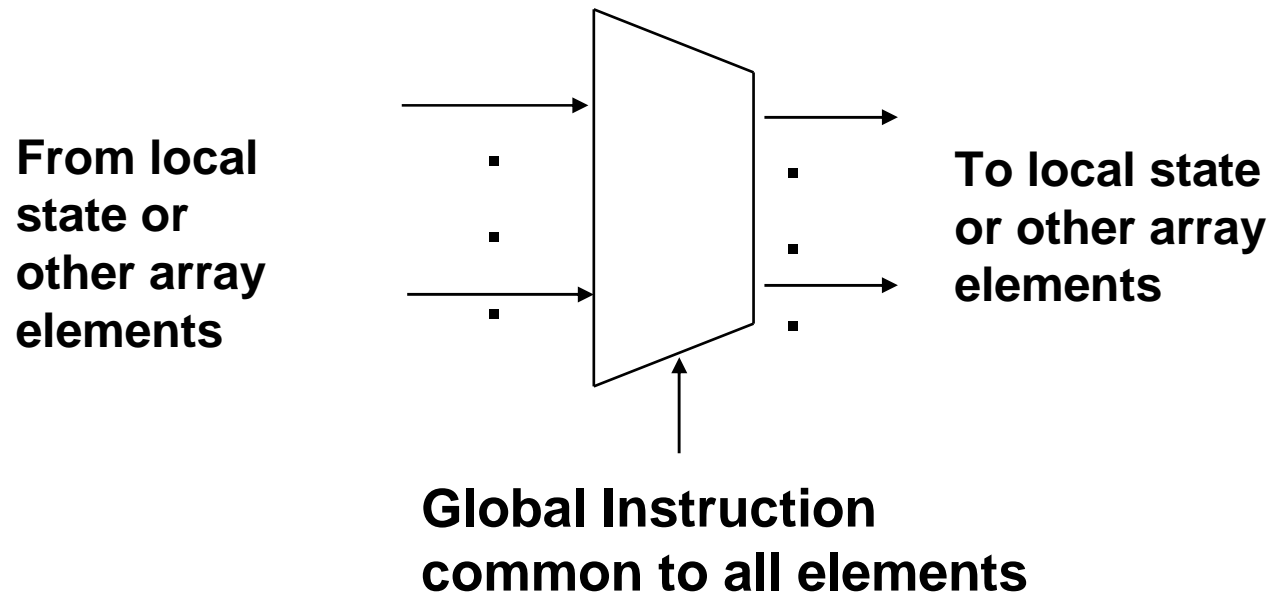


Single-Instruction Multiple Data

- Same instruction distributed to fine-grained cells.
- Typically organized as 2-D array
- Ideal for image processing
- Typically fixed hardware located in cell

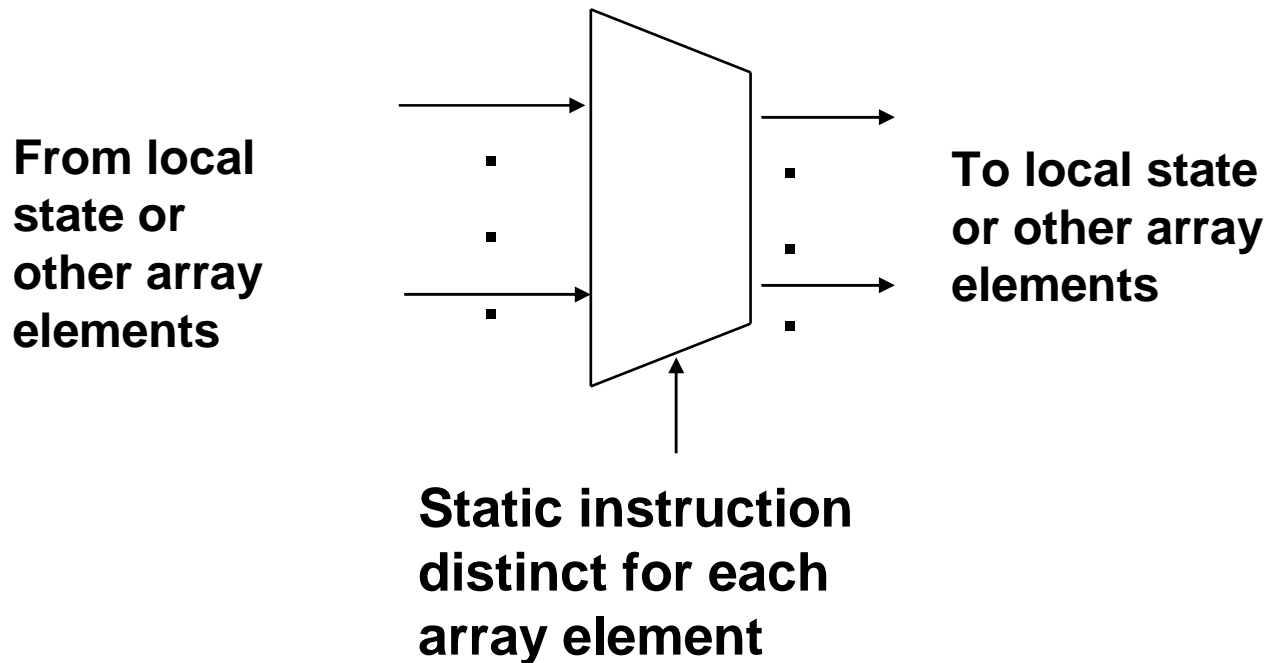


Computation Unit for SIMD



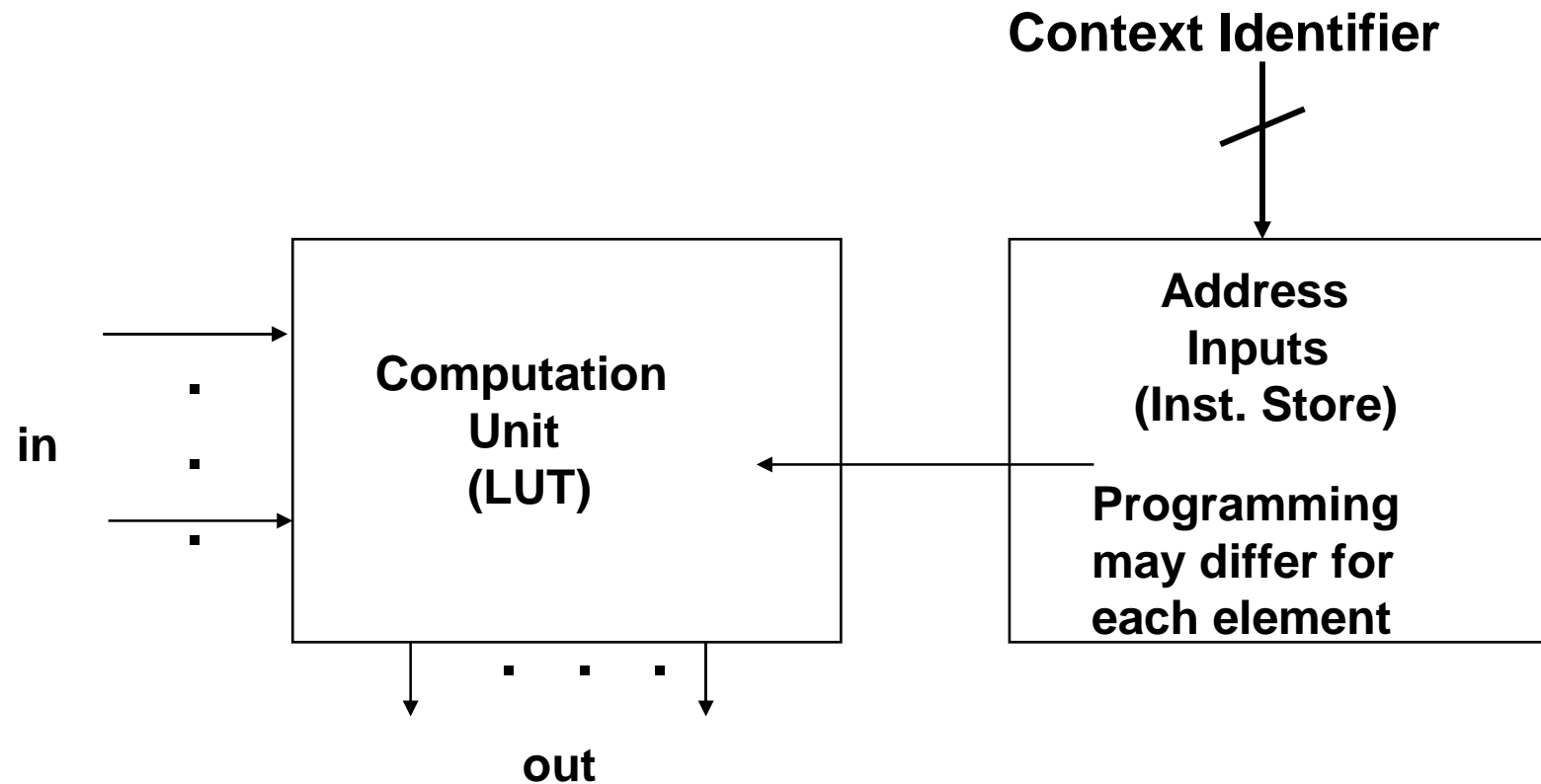
- Performs different operation on every cycle
- Easy to distribute instructions on device (use global lines)
- Some local storage for data in each tile

Computation Unit for FPGA



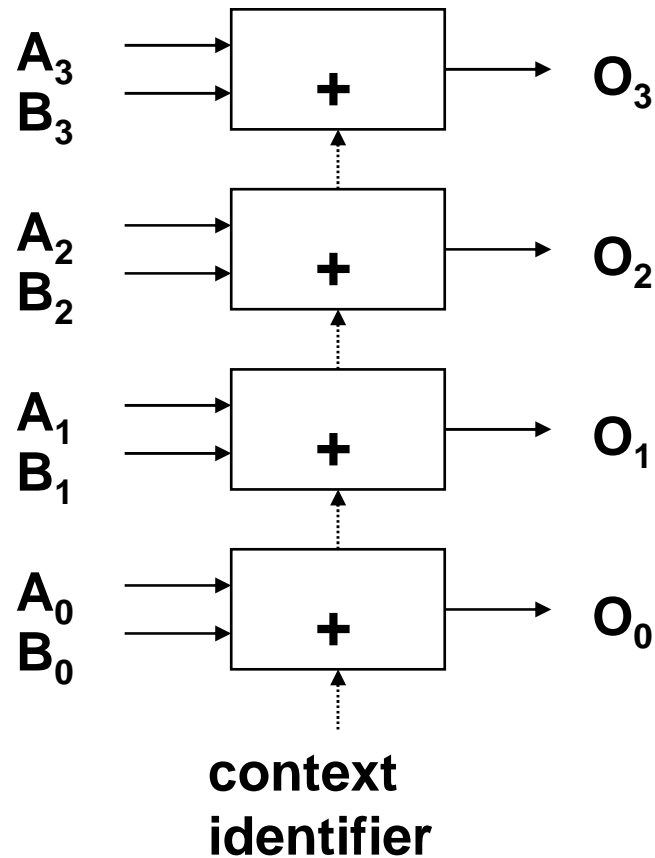
- Performs same operation on every cycle
- No global distribution of instructions at all (stored locally)
- Also has local storage for data.

Hybrid Architecture



- Configuration selects operation of computation unit
- Context identifier changes over time to allow change in functionality
- DPGA – Dynamically Programmable Gate Array

DPGA

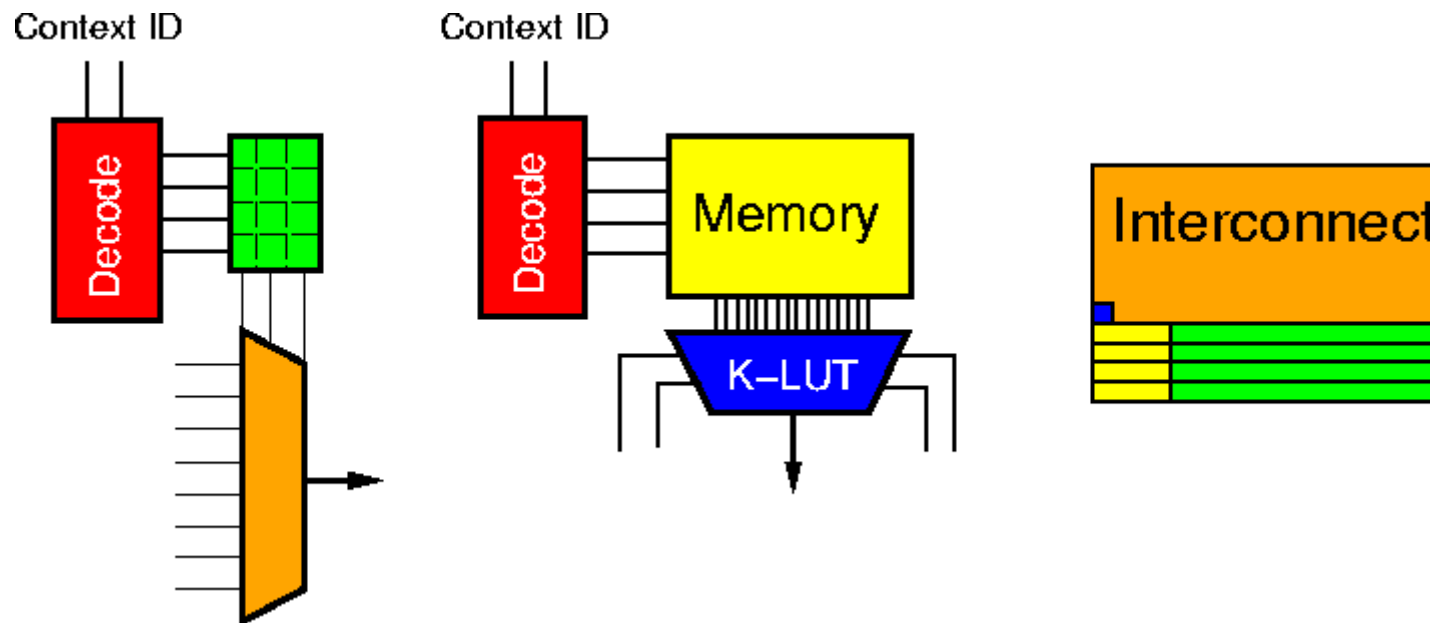


- Added configuration allows for functionality to change quickly
- Doubles SRAM storage requirement

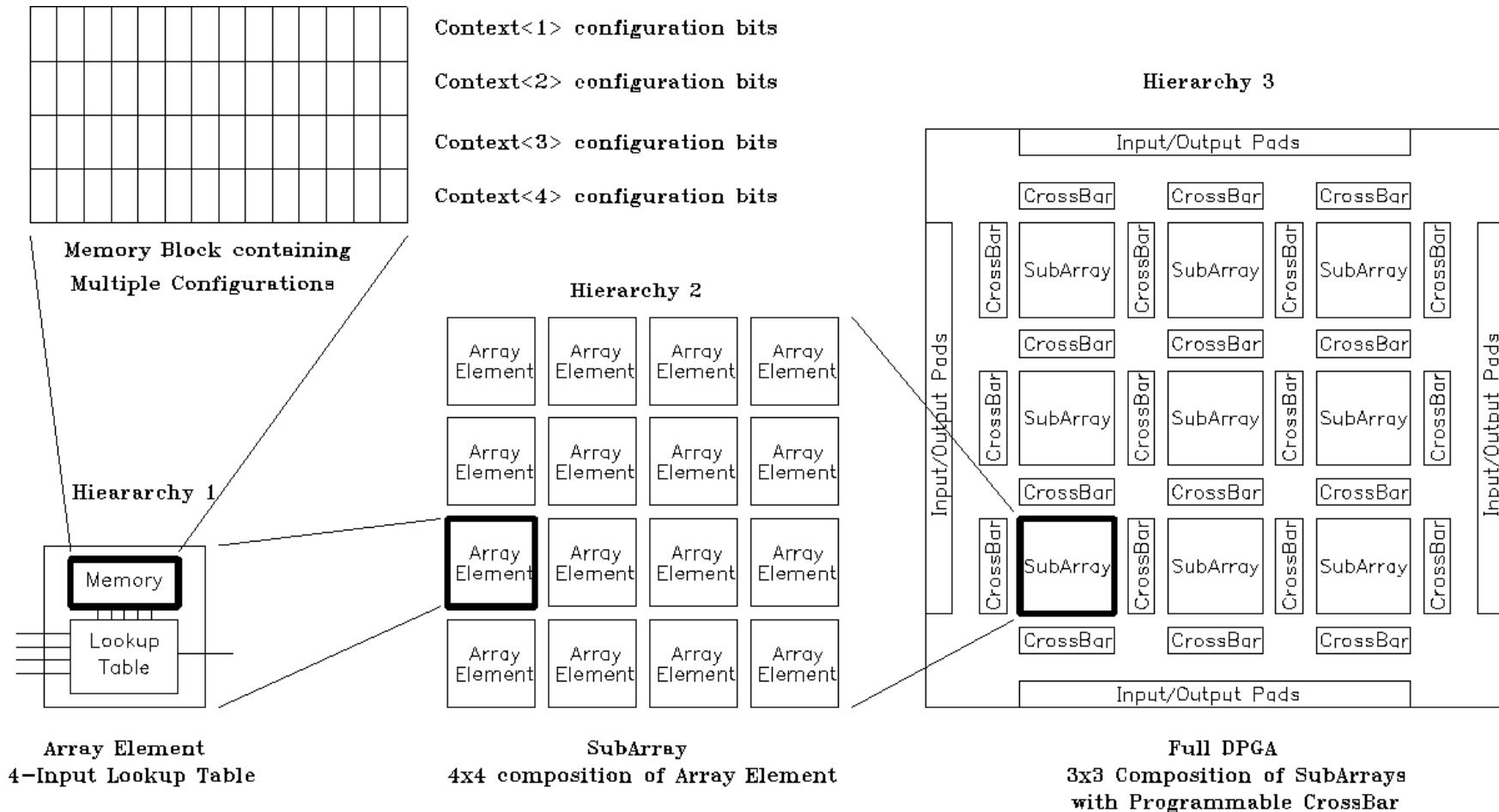
- How many applications require this flexibility
- Efficient techniques needed to schedule when functionality shifts.

Multicontext Organization/Area

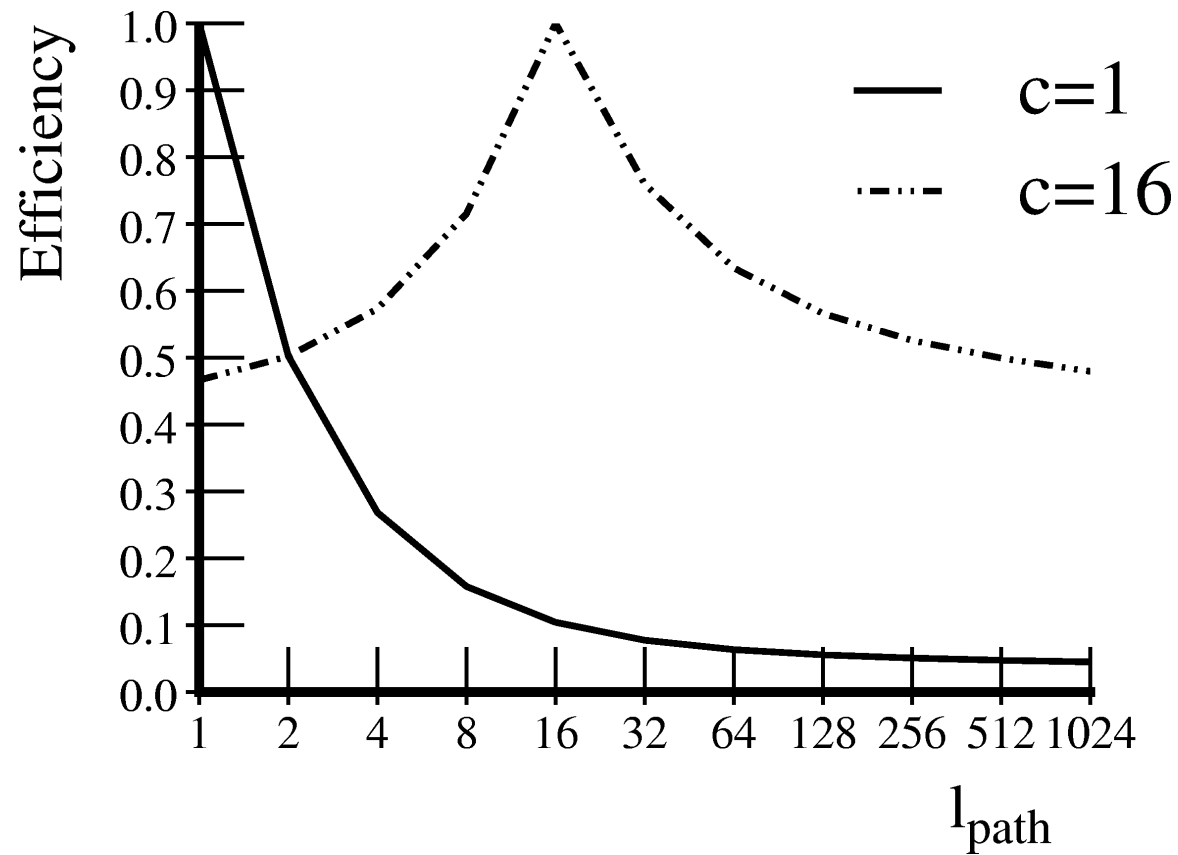
- $A_{\text{ctxt}} \approx 80K\lambda^2$
 - dense encoding
 - $A_{\text{base}} \approx 800K\lambda^2$
 - **Slides: courtesy DeHon**
- $A_{\text{ctxt}} : A_{\text{base}} = 1:10$



Example: DPGA Prototype

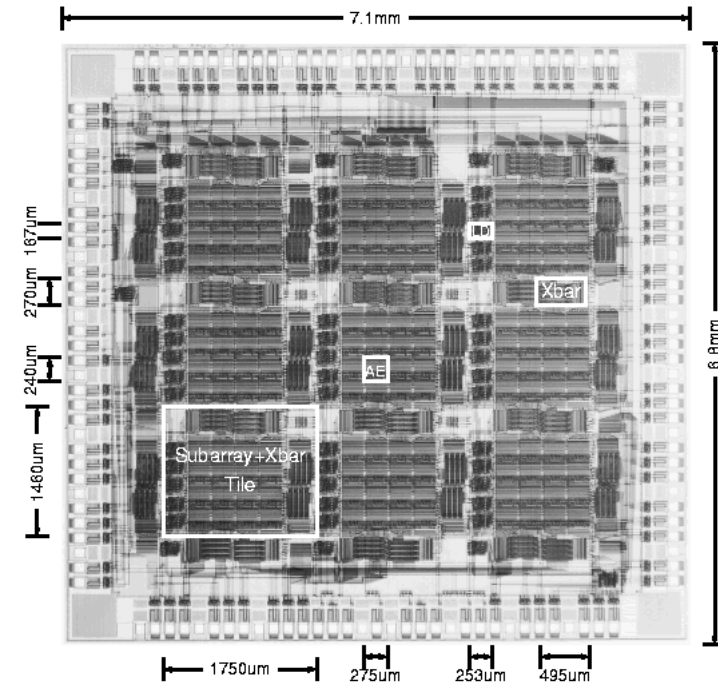


FPGA vs. DPGA Compare

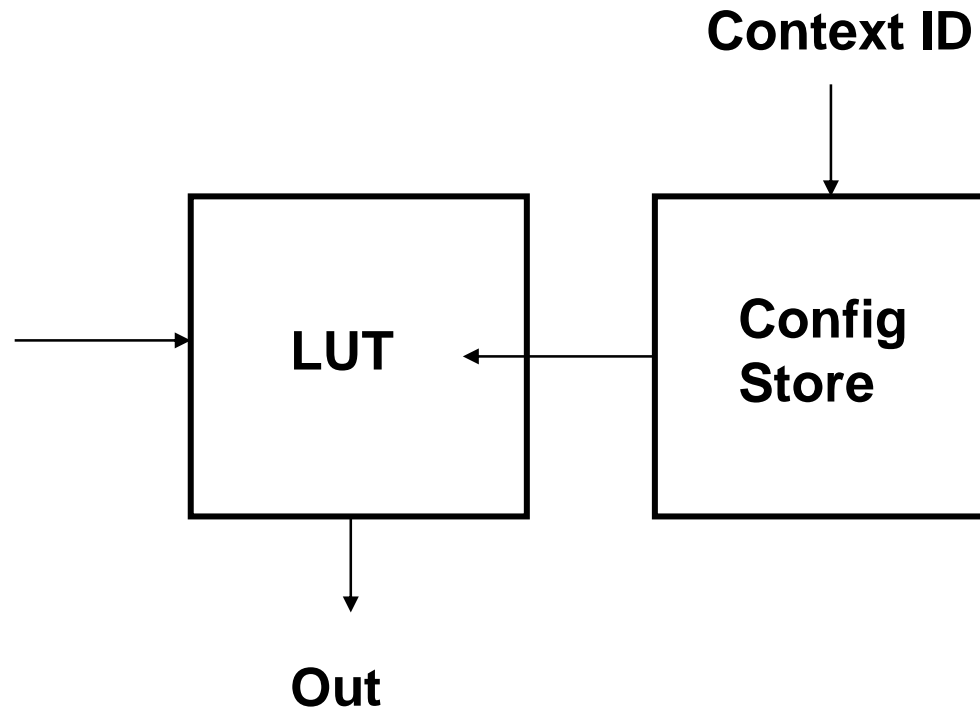


Example: DPGA Area

Process	1.0 μ CMOS
Chip	7.1mm \times 6.8mm
AEs	144
Contexts	4
AE Area	640K λ^2
A_{base}	544K λ^2
A_{ctx}	24K λ^2
$A_{base} : A_{ctx}$	20+:1
(nominal delay)	9ns

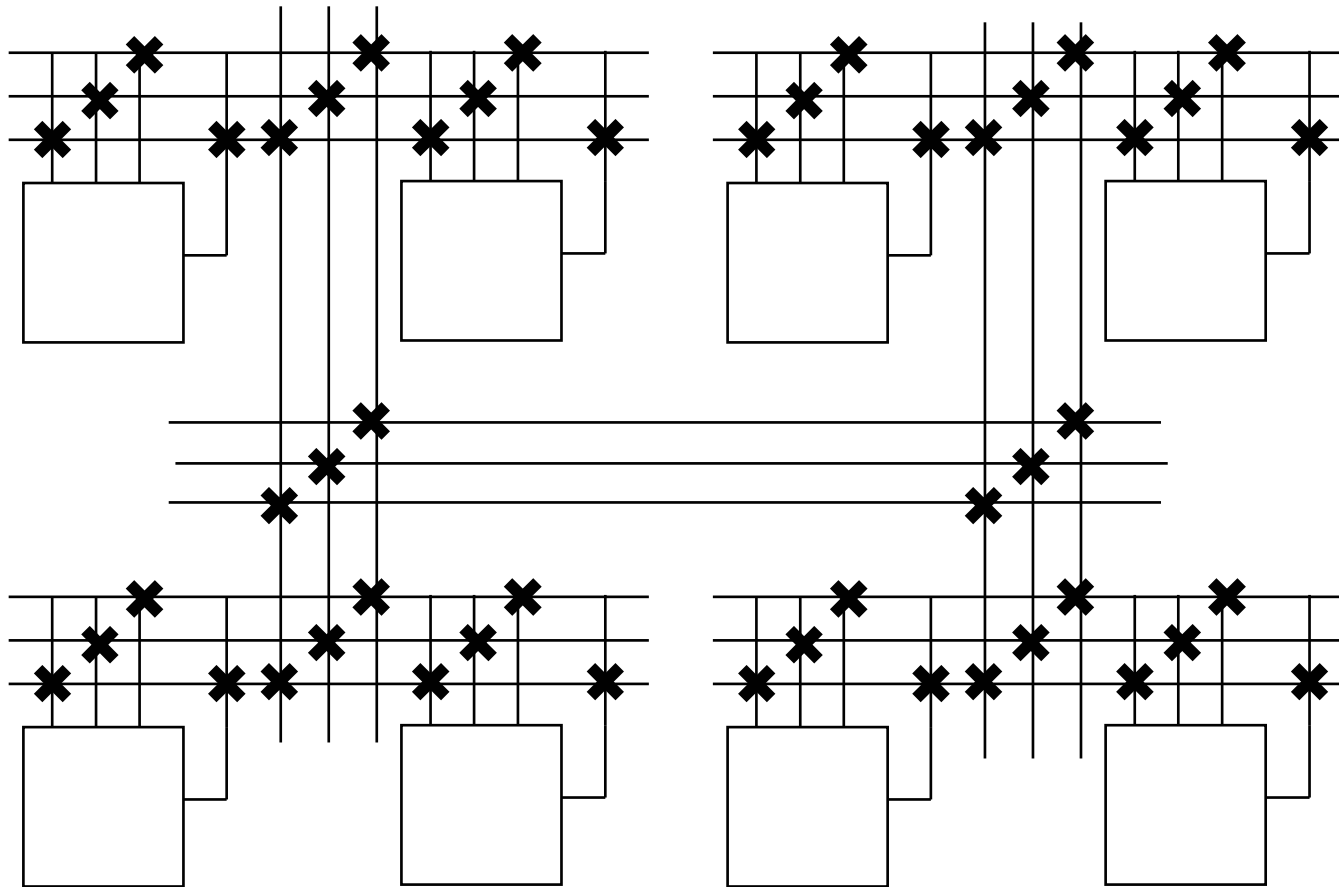


Configuration Caching



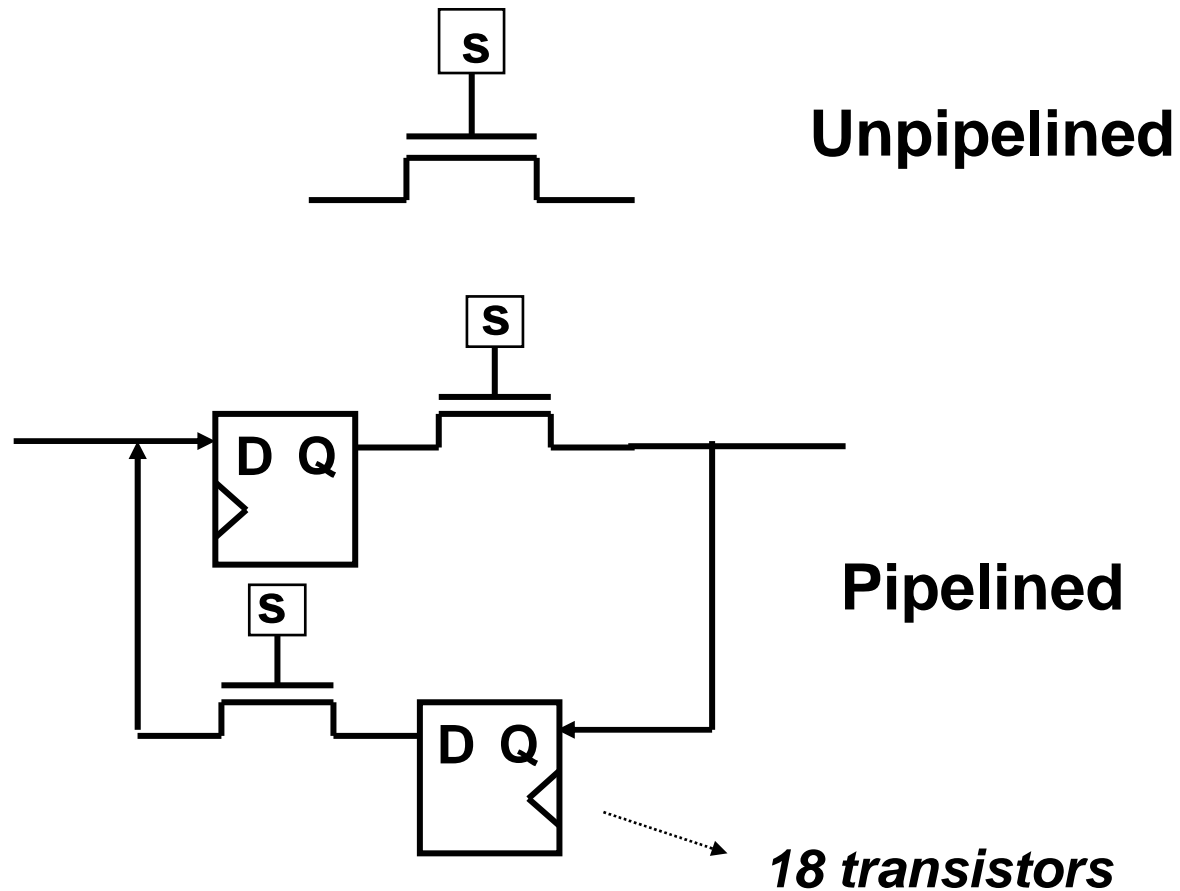
- **What if I swap out some unused configurations while they are not used?**
- **Separate hardware to write given locations in hardware (config mem) and not interrupt circuit operation**
- **Just like cache prefetching**

Hierarchical FPGA



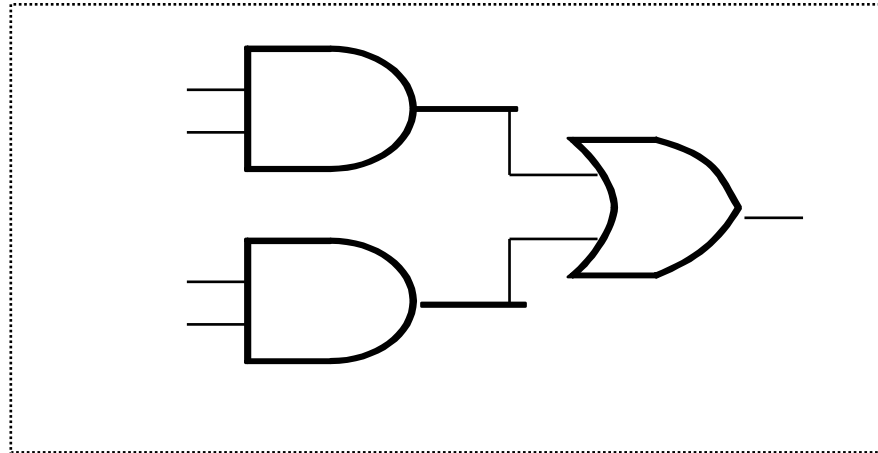
- Predictable Delay
- Two dimensional layout
- Limited connectivity

Buffering

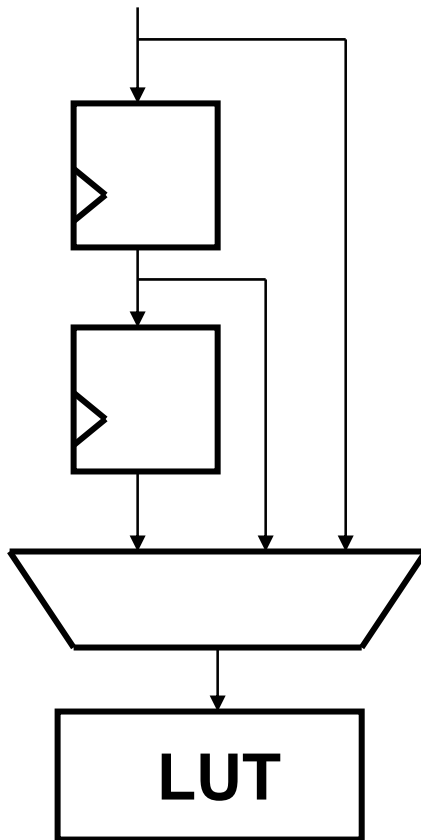


- Pipelining interconnect comes at an area cost
- Also could consider buffering

What about this circuit?



- Retiming needed for hierarchical device.
- Number of registers proportional to longest path.



Complicates design
Software, debugging

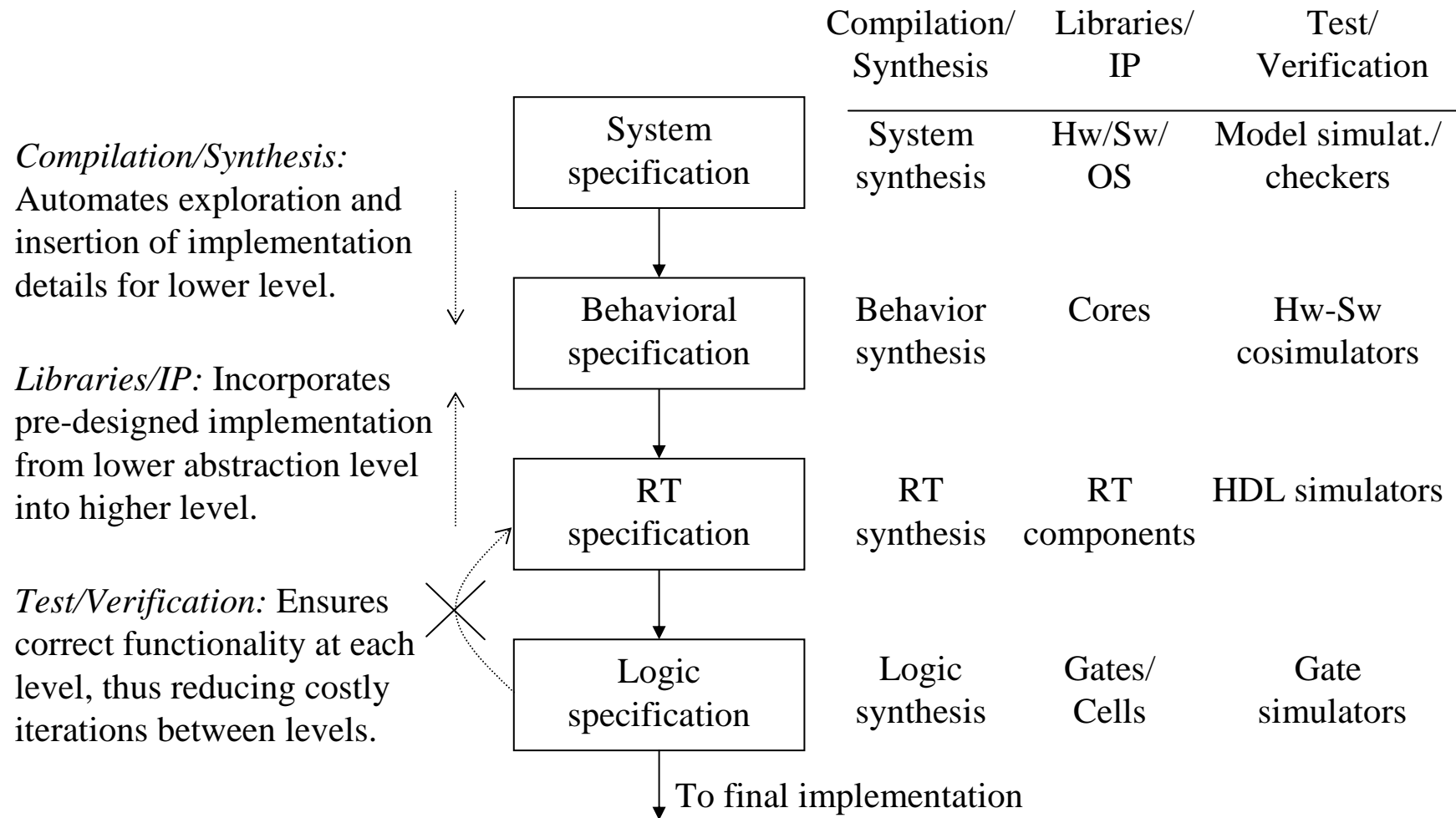
Need to schedule
communication

PLD (Programmable Logic Device)

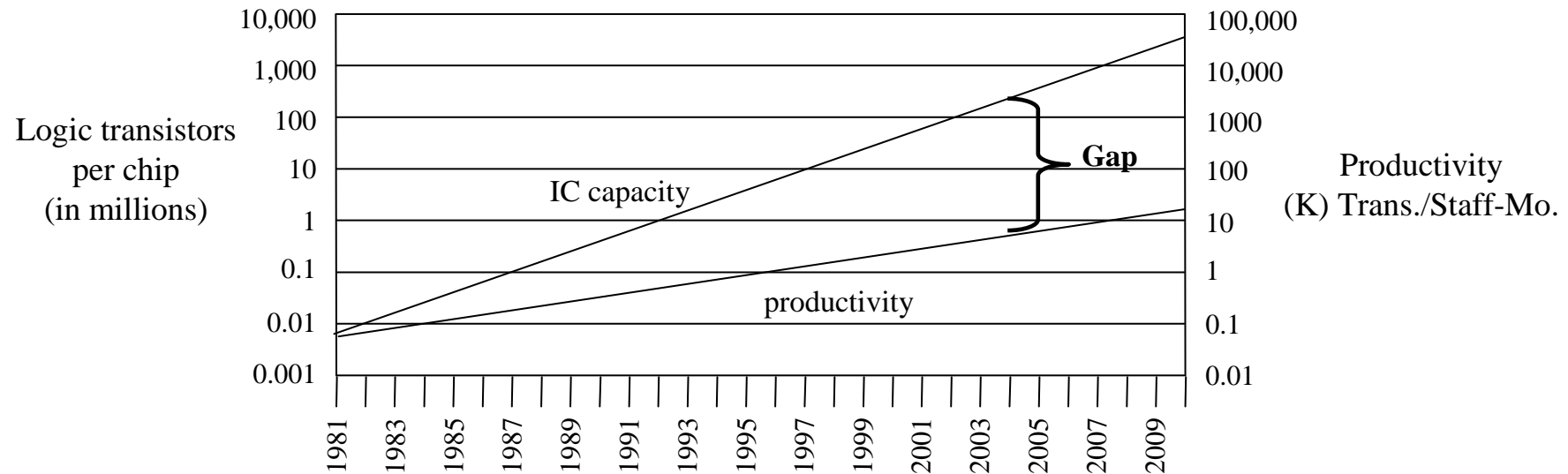
- **All layers already exist**
 - Designers can purchase an IC
 - Connections on the IC are either created or destroyed to implement desired functionality
 - Field-Programmable Gate Array (FPGA) very popular
- **Benefits**
 - Low NRE costs, almost instant IC availability
- **Drawbacks**
 - Penalty on area, cost (perhaps \$30 per unit), performance, and power
- **Acknowledgement: Mishra**

Design Technology

- The manner in which we convert our concept of desired system functionality into an implementation



Design productivity gap

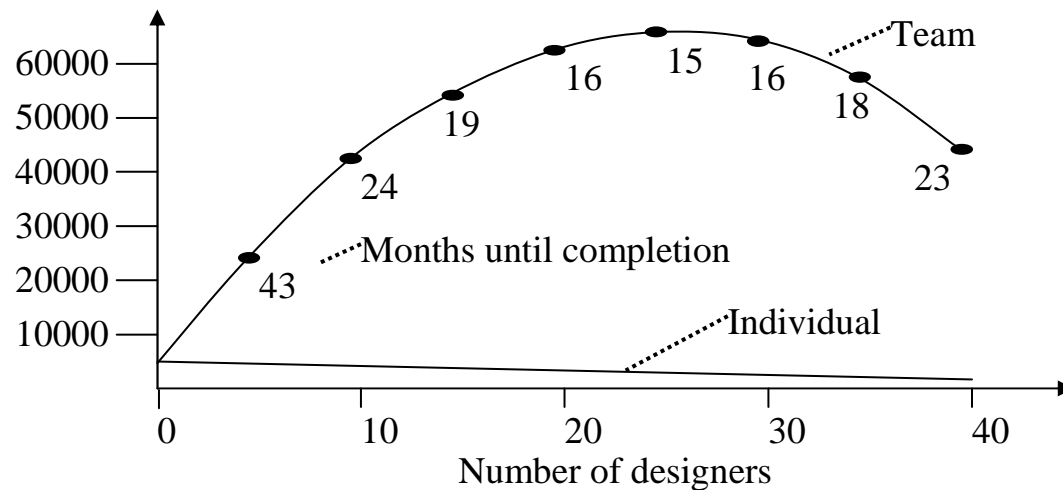


- **1981 leading edge chip required 100 man-months**
 - 10,000 transistors / 100 transistors/month
- **2002 leading edge chip requires 30K man-months**
 - 150,000,000 / 5000 transistors/month
- **Designer cost increase from \$1M to \$300M**

The mythical man-month

- In theory, adding designers to team reduces project completion time
- In reality, productivity per designer decreases due to complexities of team management and communication overhead
- In the software community, known as “the mythical man-month” (Brooks 1975)
- At some point, can actually lengthen project completion time!

- 1M transistors, one designer=5000 trans/month
- Each additional designer reduces for 100 trans/month
- So 2 designers produce 4900 trans/month each



Summary

- **Interesting similarities between processor and reconfigurable device**
- **Processors are reconfigured on every clock cycle using an instruction**
- **FPGAs configured once at beginning of computation**
- **DPGAs blur the line – run-time reconfiguration**
- **Numerous challenges to reconfiguration**
 - **When**
 - **How**
 - **Performance benefit?**