ECE 669

Parallel Computer Architecture

Reconfigurable Computing

What is Reconfigurable Computing?

- Computation using hardware that can adapt at the logic level to solve specific problems

° **Why is this interesting?**

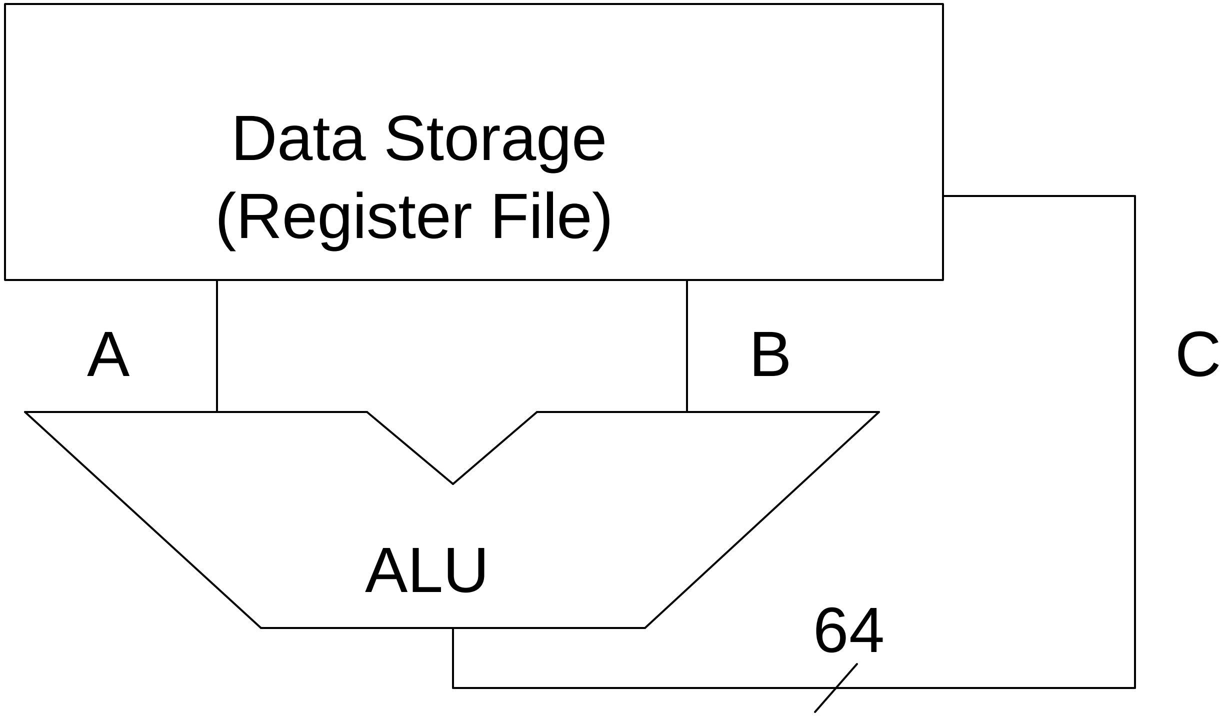
- Some applications are poorly suited to microprocessor.
- VLSI “explosion” provides increasing resources.
- Hardware/Software
- Relatively new research area.

Background needed

- Basic VLSI – transistors, delay models.
- Basic algorithms – graph algorithms, seaches
- Computer Architecture – ALU, microprocessor
- Digital Design – adder, counter, etc.

Topic self-contained!

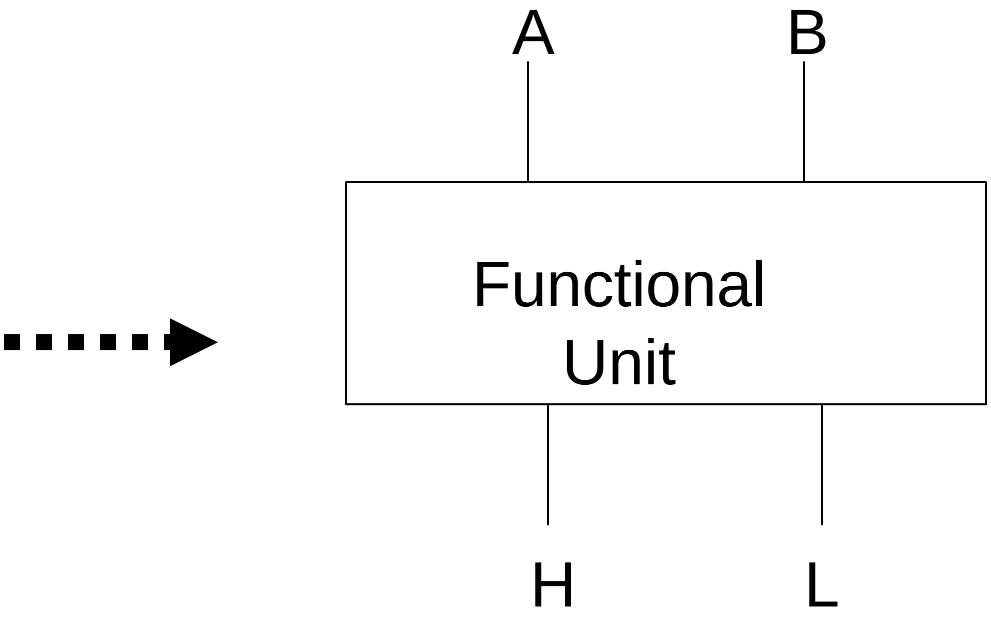
Microprocessor-based Systems



- **Generalized to perform many functions well.**
- **Operates on fixed data sizes.**
- **Inherently sequential.**

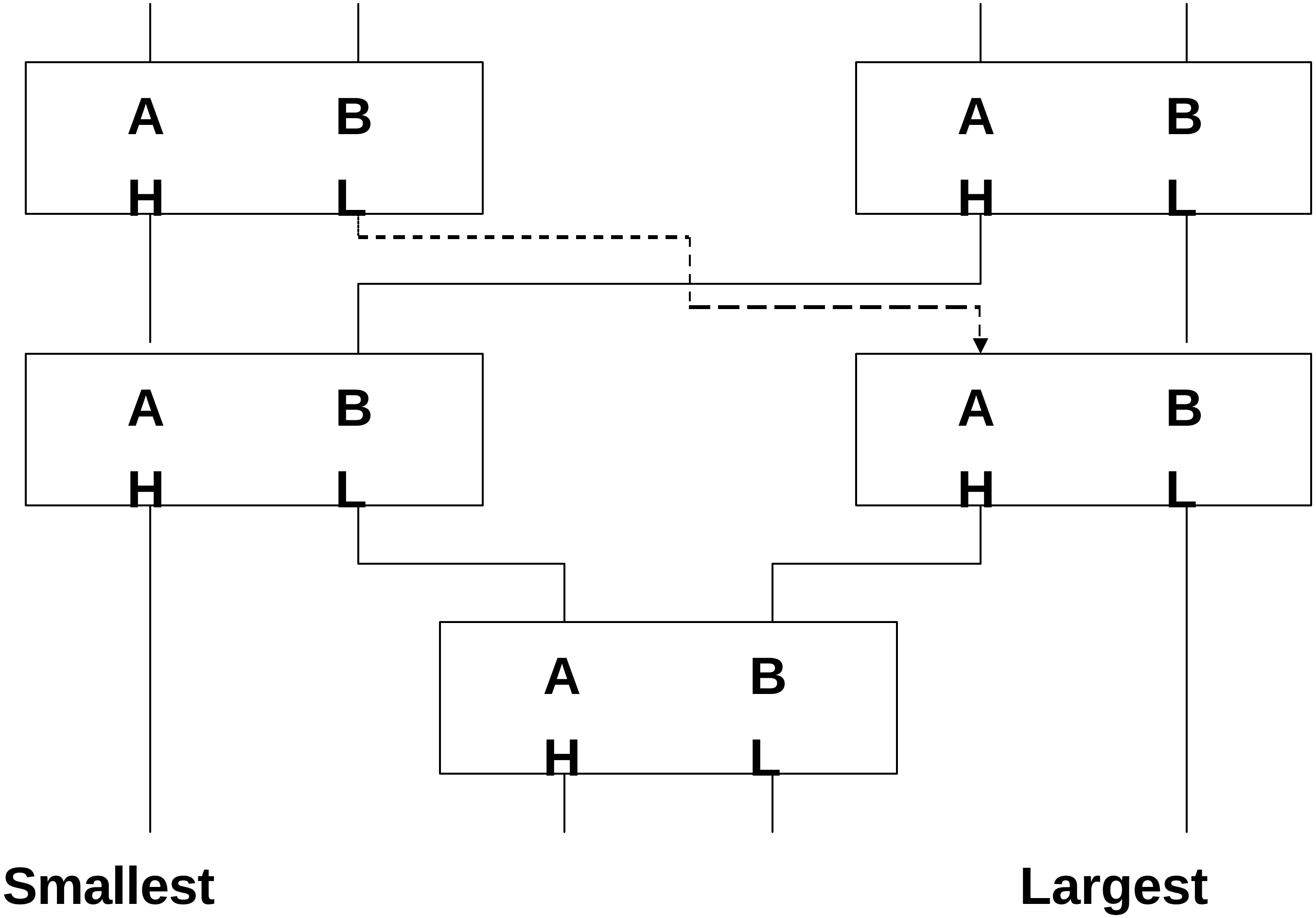
Reconfigurable Computing

```
If (A > B) {
  H = A;
  L = B;
}
Else {
  H = B;
  L = A;
}
```



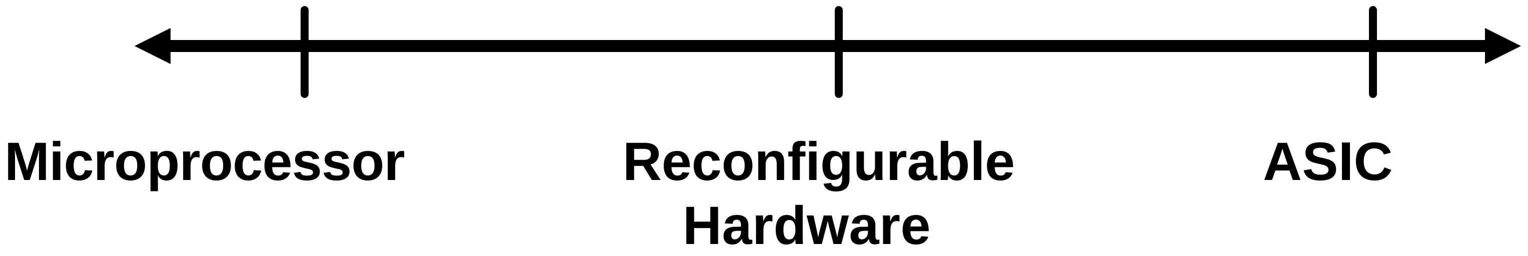
- **Create specialized hardware for each application.**
- **Functional units optimized to perform a special task.**

Example: Bubblesort



- Adapt interconnect to problem.
- Take advantage of parallelism.

Implementation Spectrum



Microprocessor

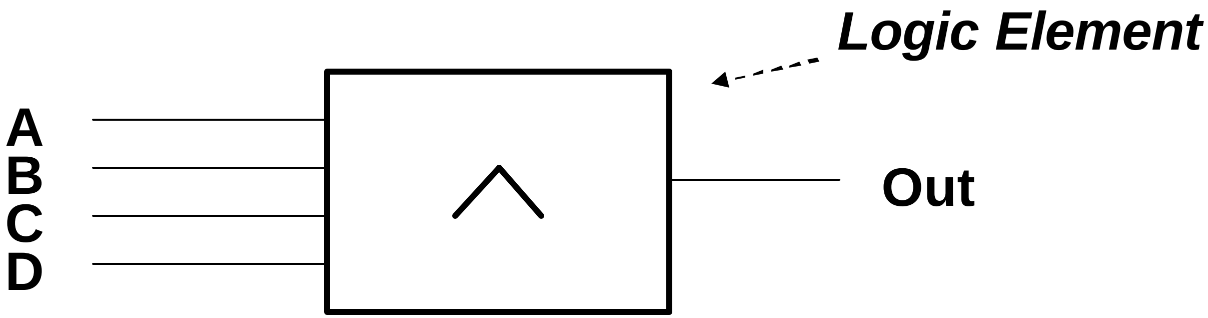
Reconfigurable Hardware

ASIC

- ASIC gives high performance at cost of inflexibility.
- Processor is very flexible but not tuned to the application.
- Reconfigurable hardware is a nice compromise.

What does it look like?

Reconfigurable Hardware

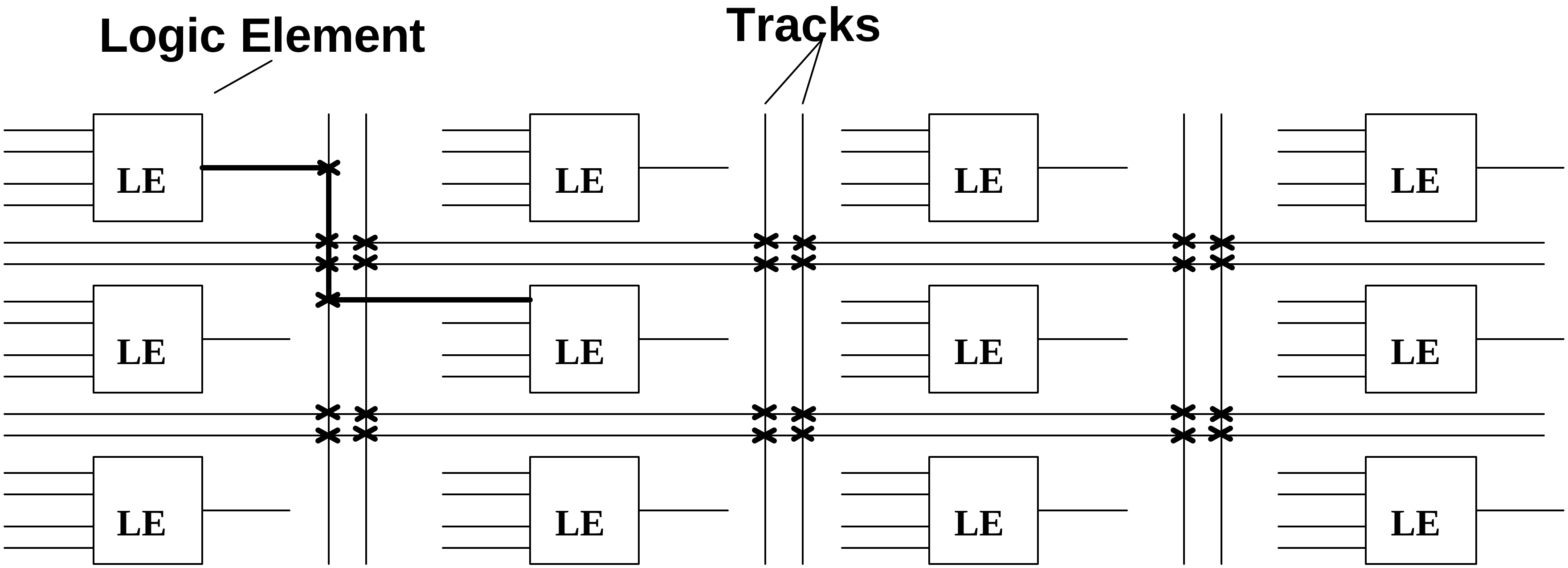


$A \wedge B \wedge C \wedge D = out$

- Each logic element operates on four one-bit inputs.
- Output is one data bit.
- Can perform <u>any</u> boolean function of four inputs

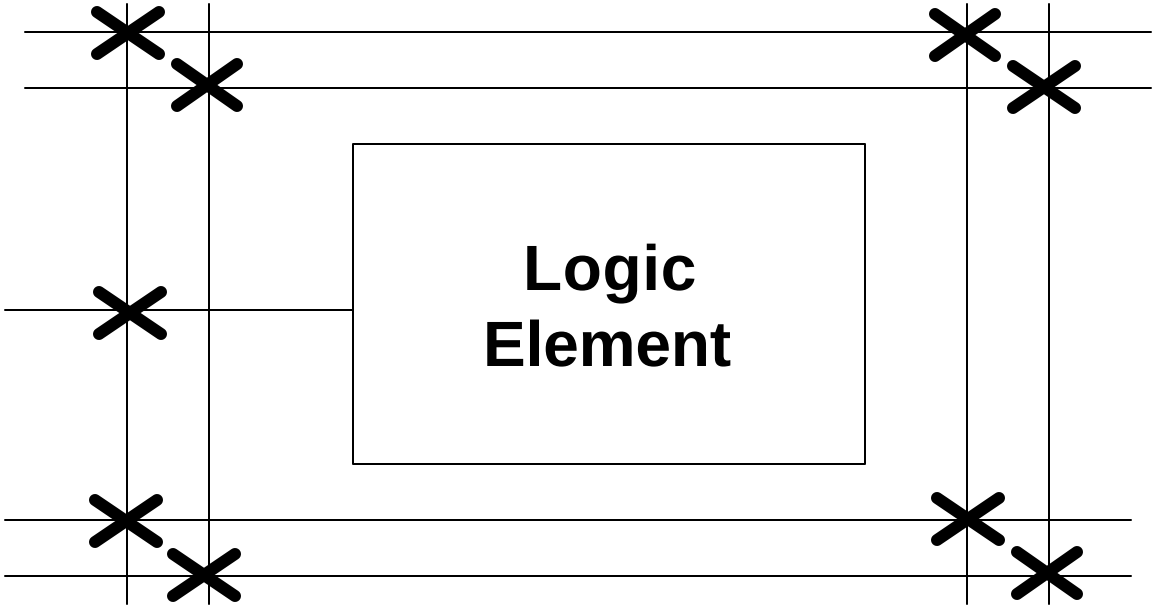
$2^{2^{4}}$ = 64K functions!

Field-Programmable Gate Array



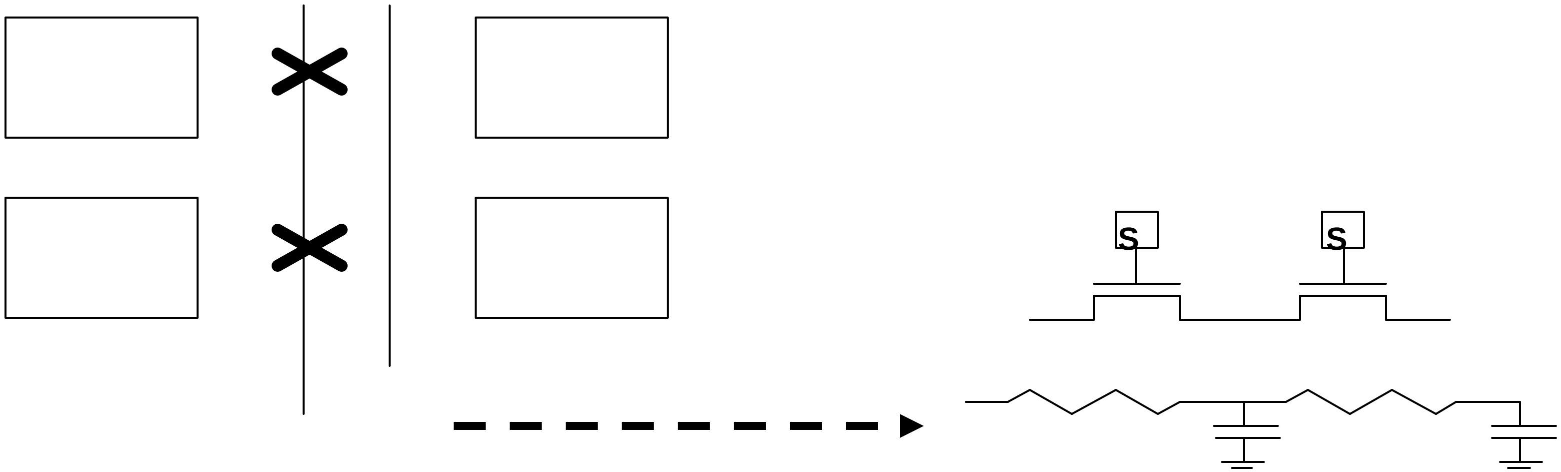
- Each *logic element* outputs one data bit.
- Interconnect programmable between elements.
- Interconnect *tracks* grouped into channels.

FPGA Architecture Issues



- Need to explore architectural issues.
- How much functionality should go in a logic element?
- How many routing tracks per channel?
- Switch “population”?

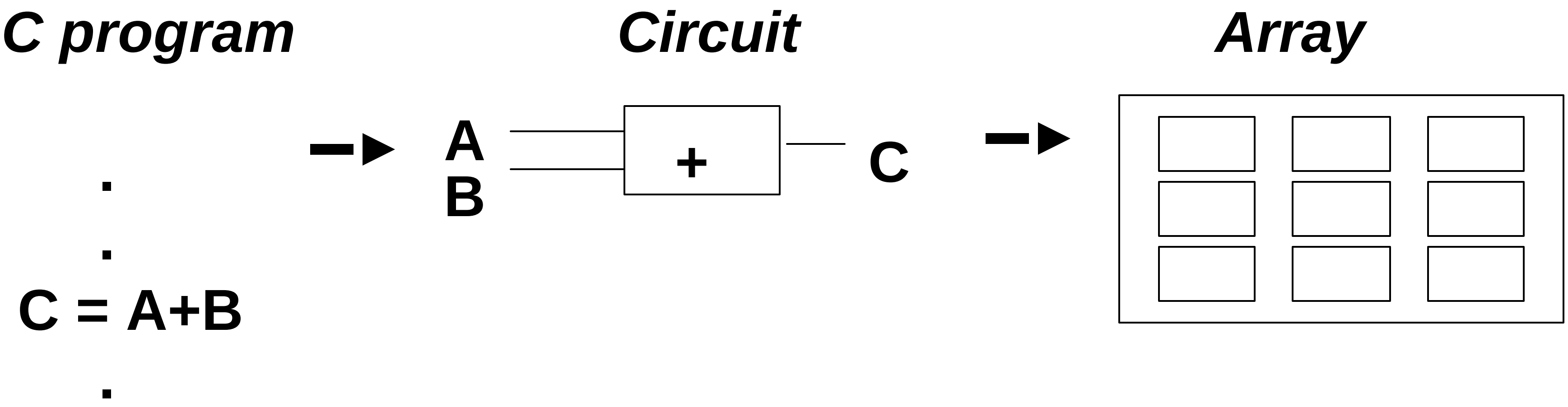
Real World Physical Issues



Wires have real cost

- Modelling FPGA delay.
- Improving performance through buffering/segmentation.
- Technology dependent.
- The cost of reconfigurability.

Translating a Design to an FPGA



C program

.

.

C = A+B

.

Circuit

A

B

+

C

Array

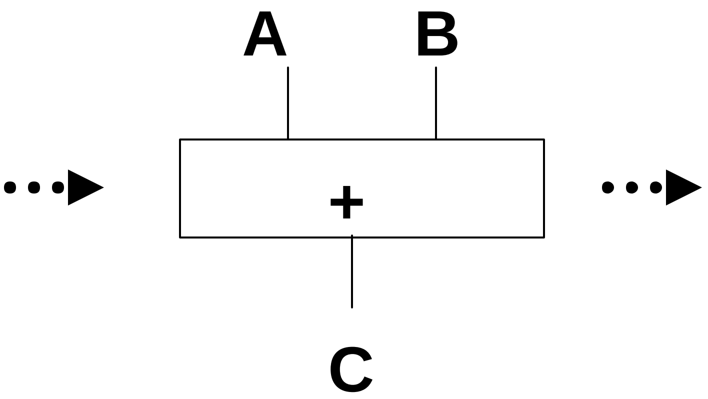
- CAD to translate circuit from text description to physical implementation well understood.
- CAD to translate from C program to circuit not well understood.
- Very difficult for application designers to successfully write high-performance applications

Need for design automation!

High-level Compilers

- **Difficult to estimate hardware resources.**
- **Some parts of program more appropriate for processor (hardware/software codesign).**
- **Compiler must parallelize computation across many resources.**
- **Engineers like to write in C rather than pushing little blocks around.**

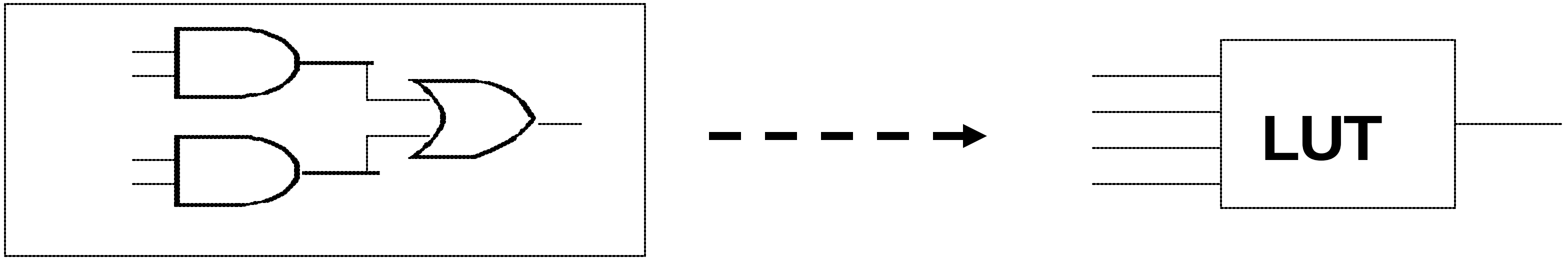
C = A+B



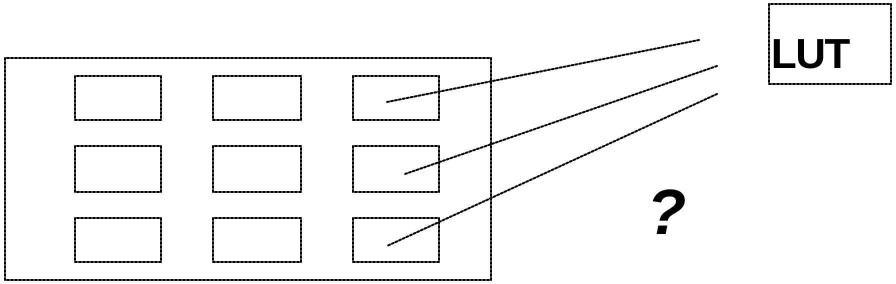
```
for (i = 0; i<n, i++)
{
    .
    .
}
```

Circuit Compilation

1. Technology Mapping

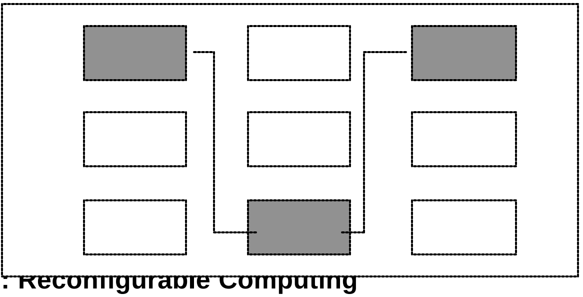


2. Placement



Assign a logical LUT to a physical location.

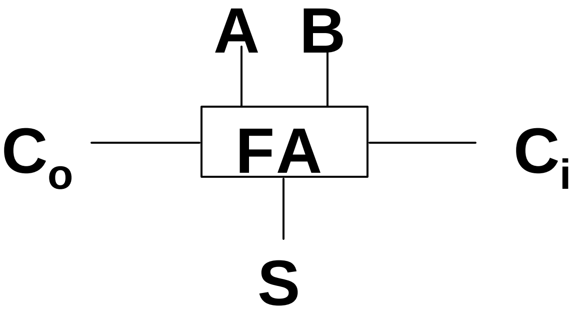
3. Routing



**Select wire segments
And switches for
Interconnection.**

Two Bit Adder

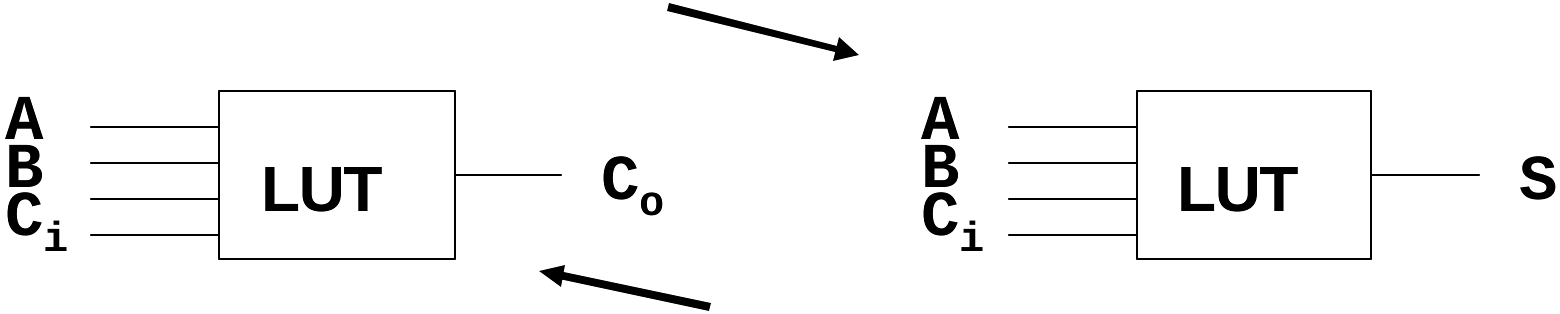
Made of Full Adders



$$A+B = D$$

Logic synthesis tool reduces circuit to SOP form

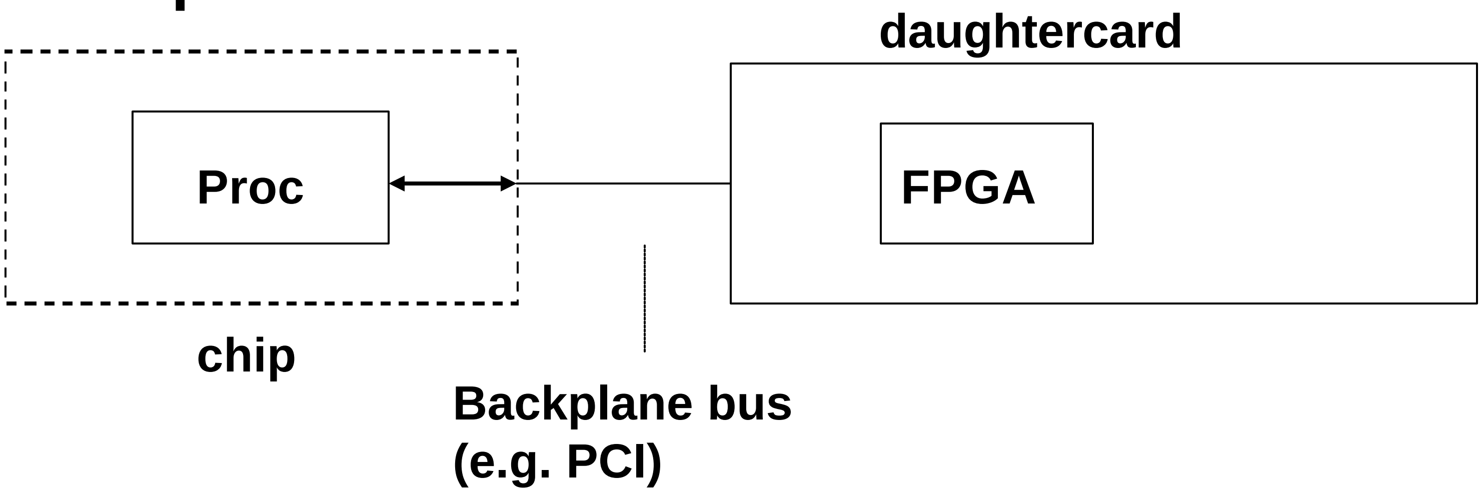
$$S = \overline{A}B\overline{C}_i + A\overline{B}\overline{C}_i + \overline{A}\overline{B}C_i + ABC_i$$



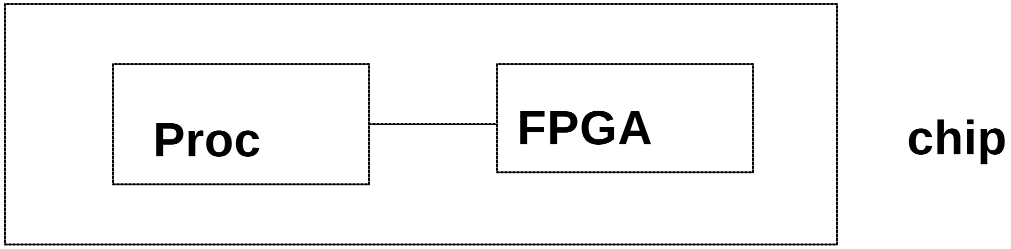
$$C_o = \overline{A}BC_i + A\overline{B}C_i + AB\overline{C}_i + ABC_i$$

Processor + FPGA

Three possibilities



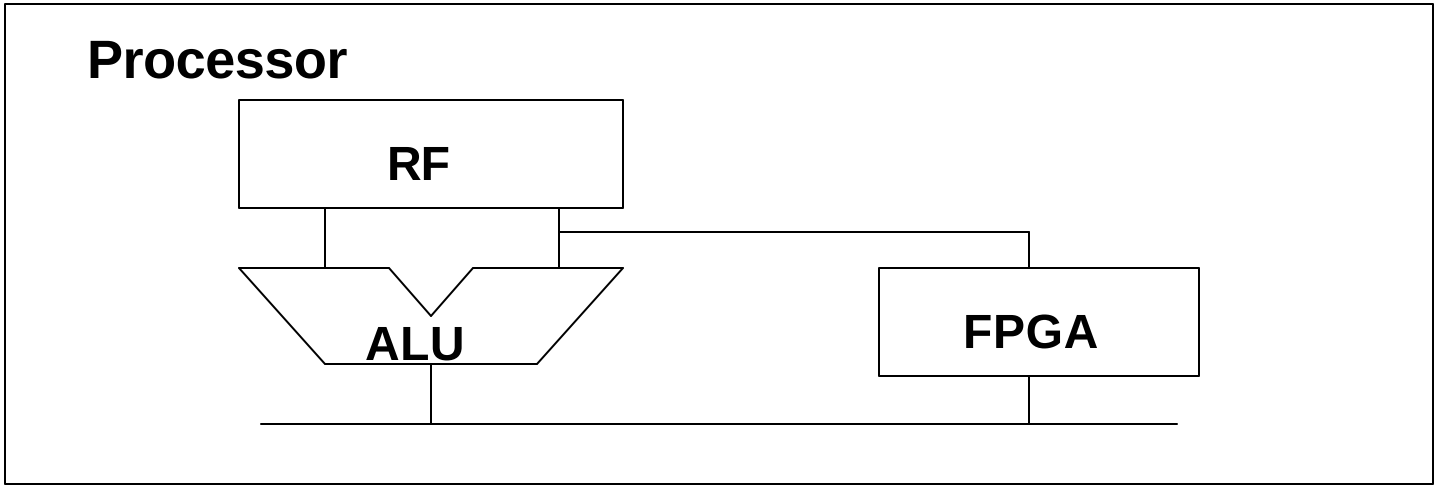
1. FPGA serves as coprocessor for data intensive applications – possible project.



2. FPGA serves as embedded computer for low latency transfer. *“Reconfigurable Functional Unit”*

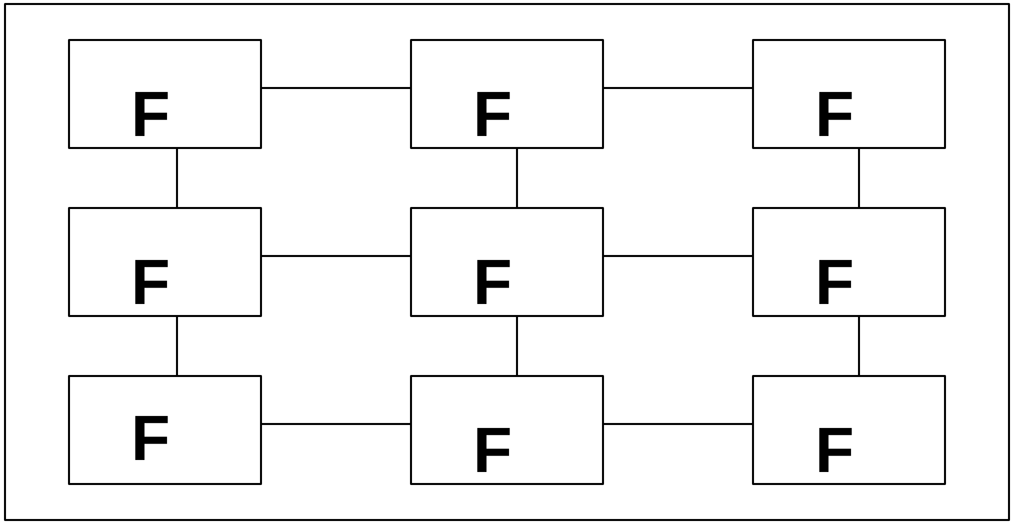
Processor + FPGA (cont..)

3. Processor integration



- FPGA logic embedded inside processor.
- A number of problems with 2 and 3.
  - Process technology an issue.
  - ALU much faster than FPGA generally.
  - FPGA much faster than the <u>entire</u> processor.

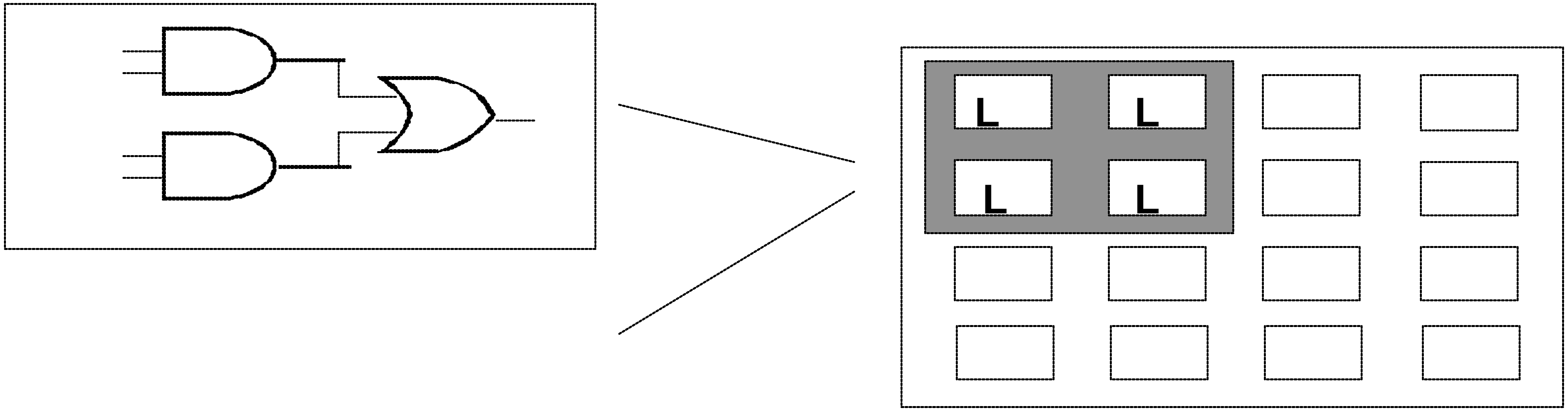
Multi-FPGA Systems



- Most applications don’t fit on one device.
- Create need for partitioning designs across many devices.
- Effectively a “netlist computer”

Each FPGA is a logic processor interconnected in a given topology.

Dynamic Reconfiguration



- What if I want to exchange part of the design in the device with another piece?
- Need to create architectures and software to incrementally change designs.
- Effectively a “configuration cache”

Examples: encryption, filtering.

Research Areas

- Storing configuration info inside device.
- Architecture evaluation.
  - Size and performance tradeoff.
- Layout of a new logic element.
- Algorithm for place and route.
- Apply an application to FPGA logic.

Versatile Place and Route

- Written by Vaughn Betz at the University of Toronto
- Performs FPGA placement and routing.
- Written in C
- Runs on Suns, Alphas, Linux
- Estimates device sizes and performance.

Xilinx XC4000 Cell

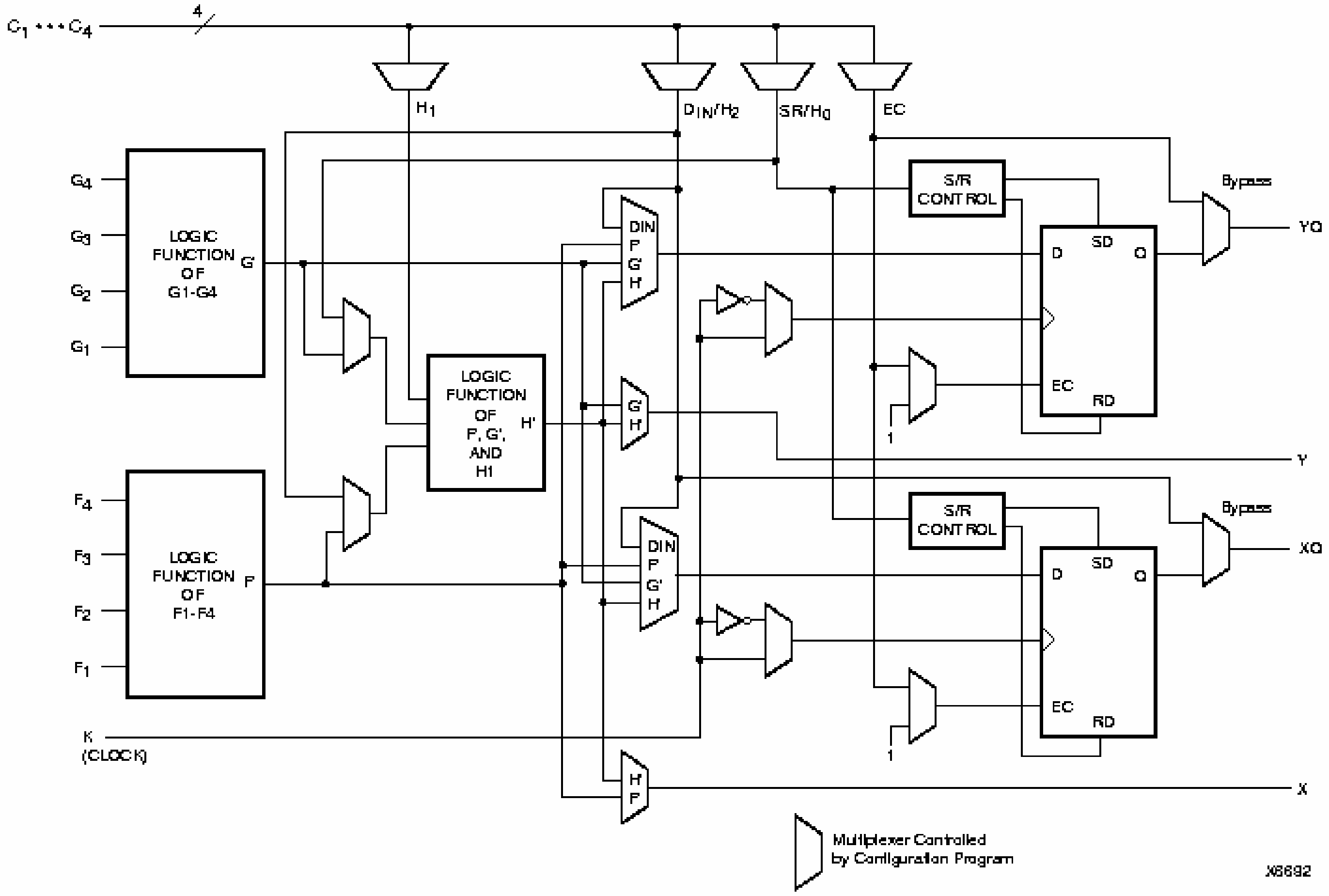
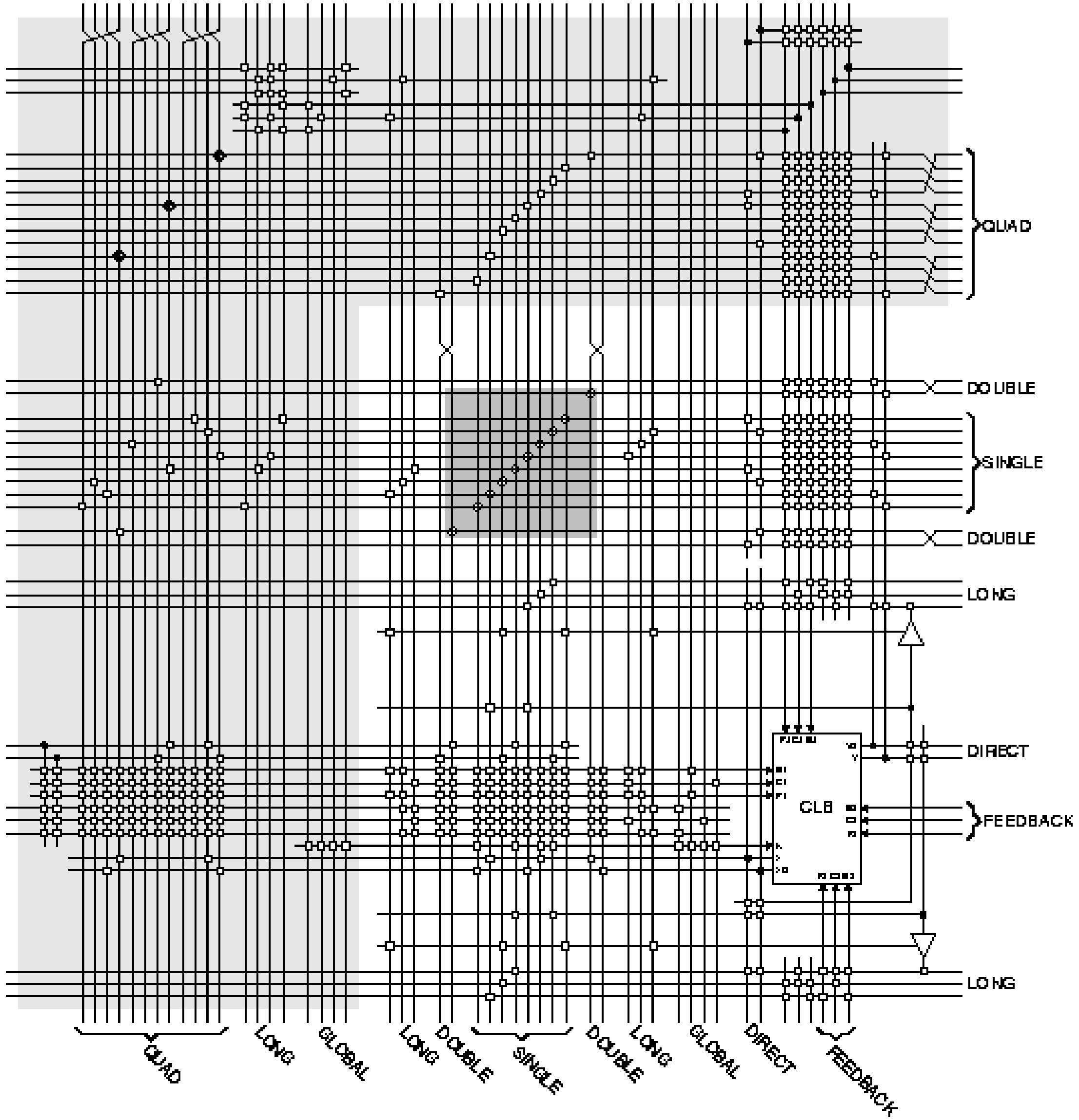


Figure 1: Simplified Block Diagram of XC4000 Series CLB (RAM and Carry Logic functions not shown)

- 2 4-input look-up tables
- 1 3-input look-up table
- 2 D flip flops

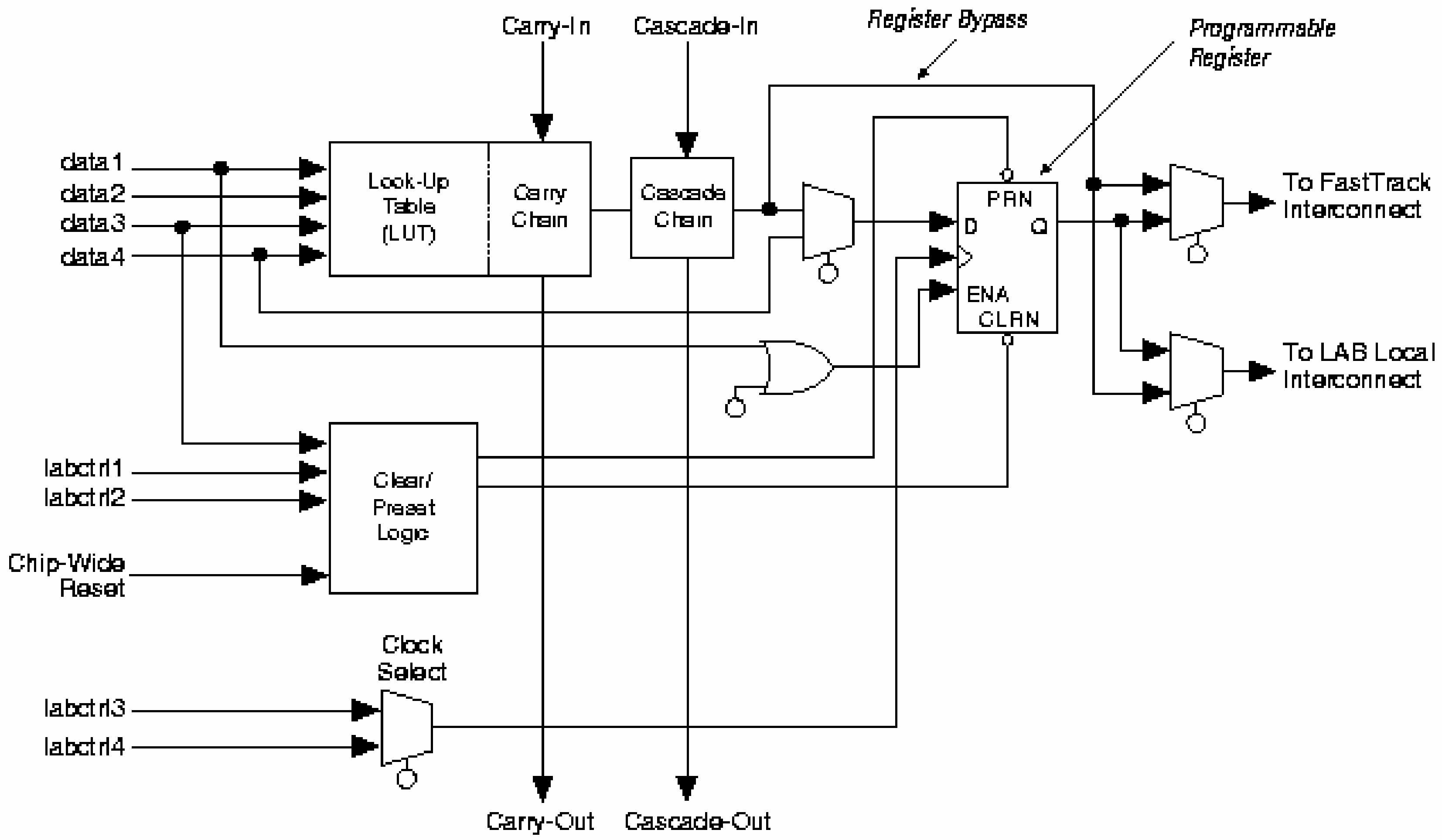
Xilinx XC4000 Routing

XC4000E and XC4000X Series Field Programmable Gate Arrays

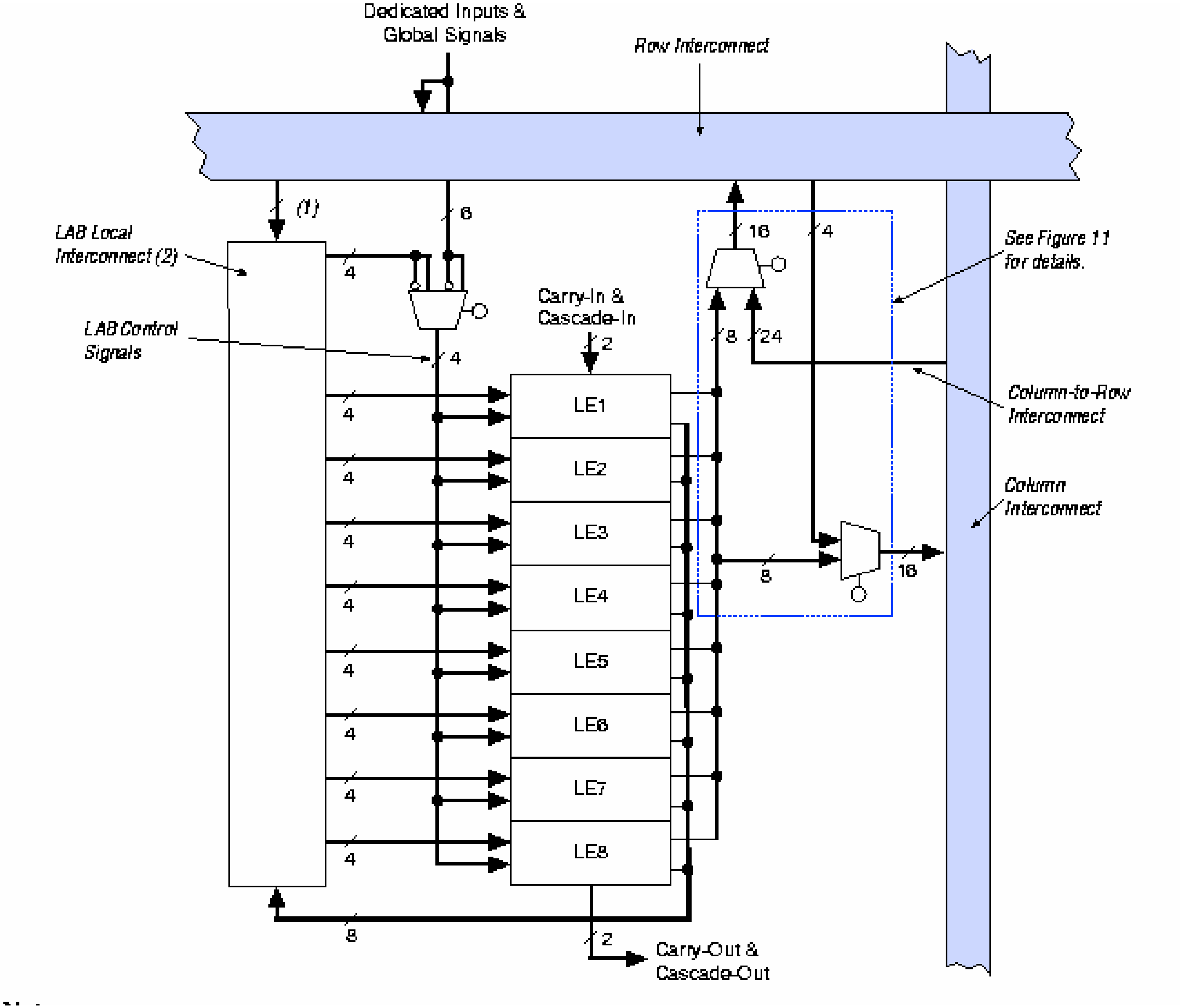


Altera Flex10K

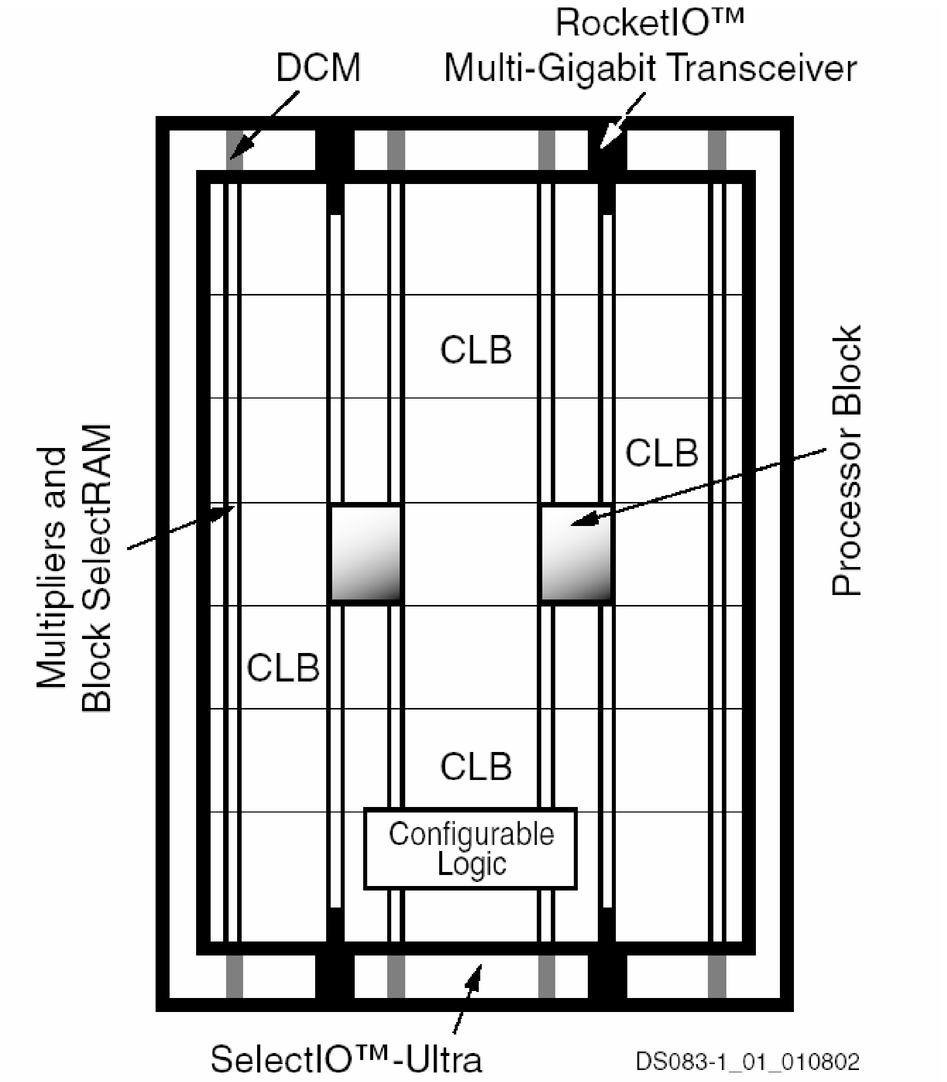
Figure 6. FLEX 10K Logic Element



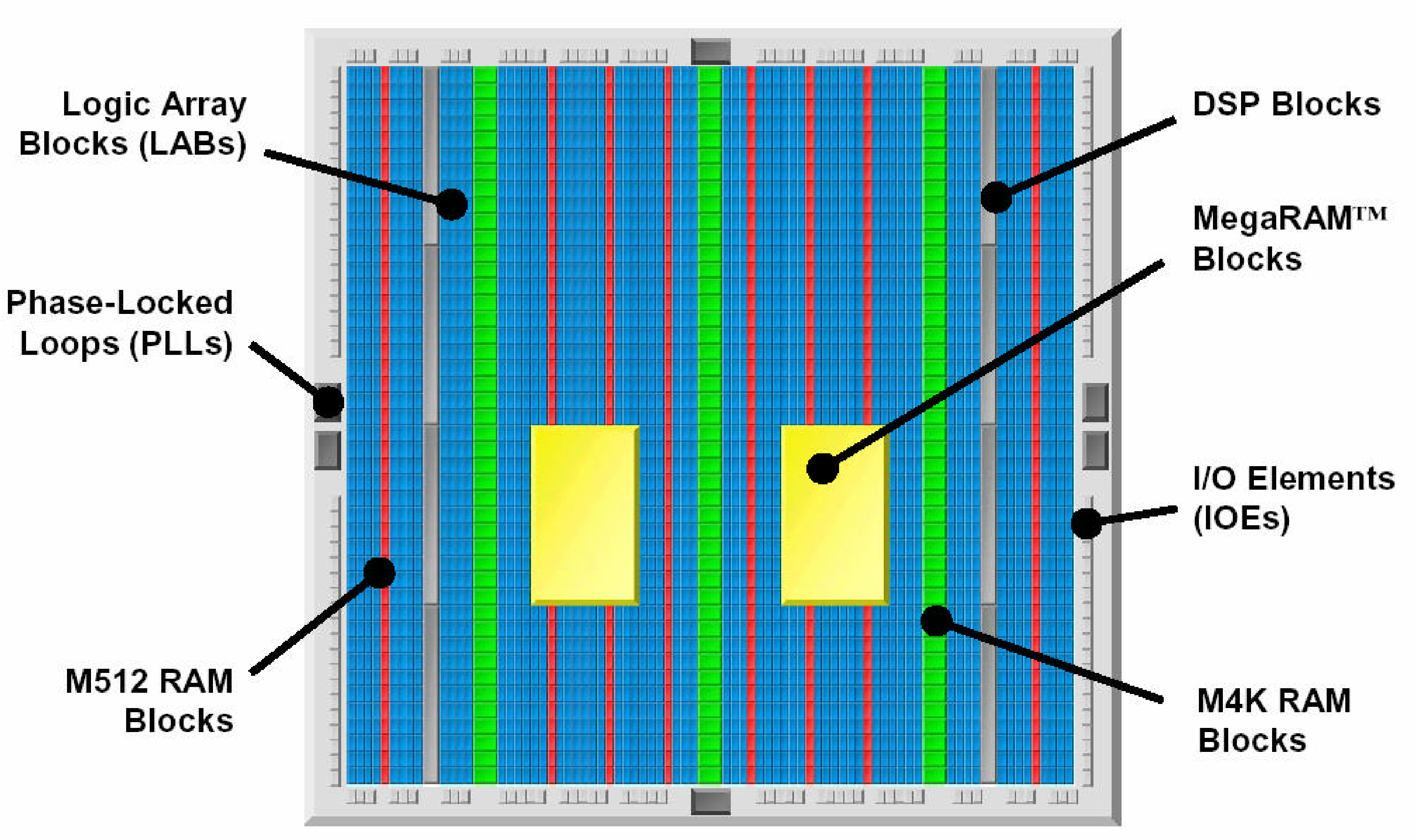
Altera Flex10K



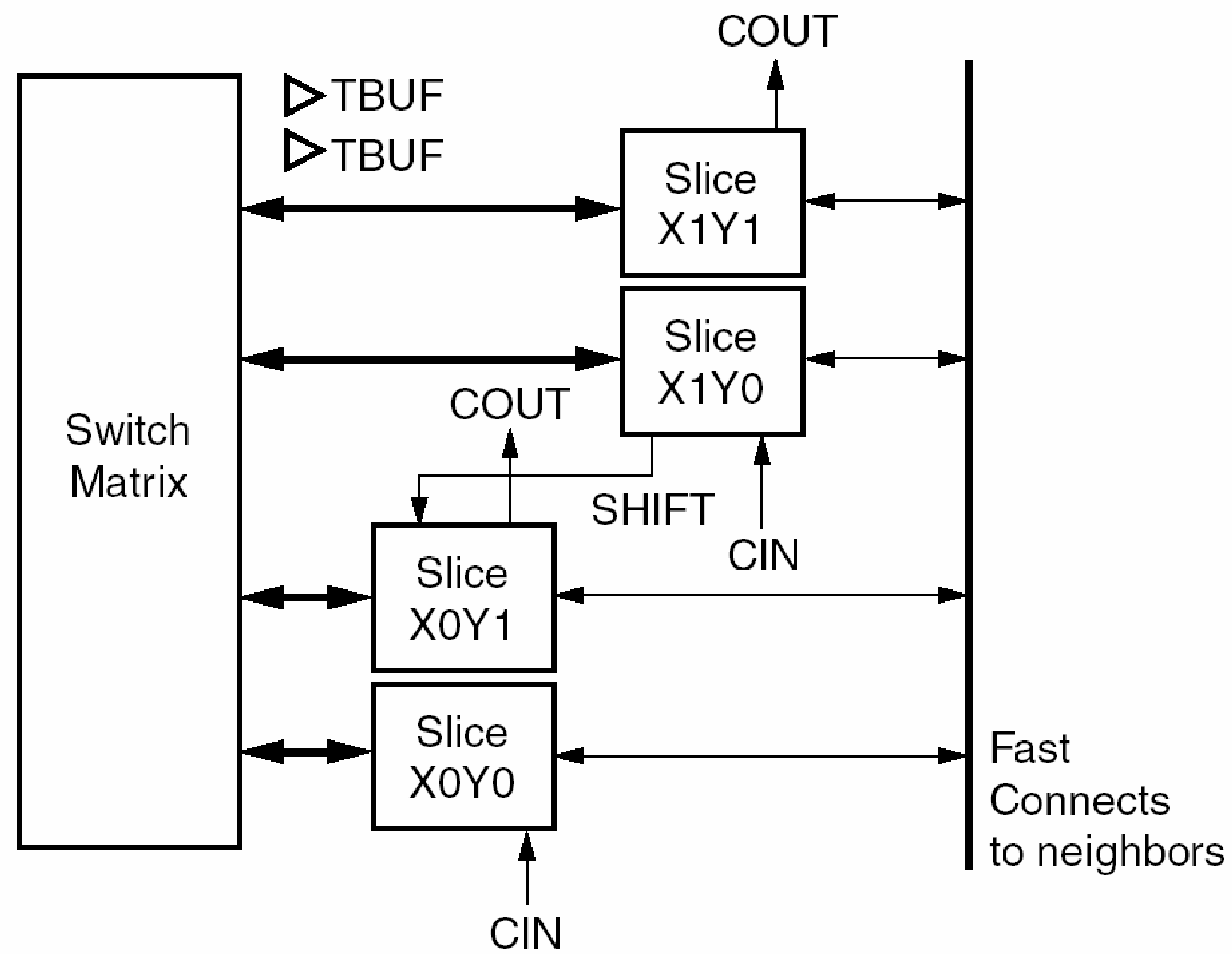
Xilinx Virtex-II Pro



Altera Stratix

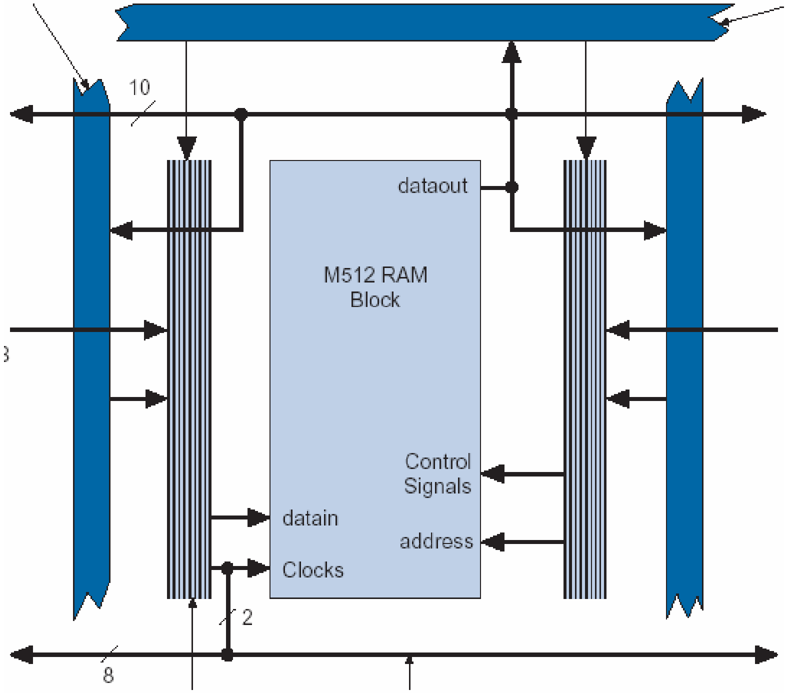


Xilinx Virtex CLB

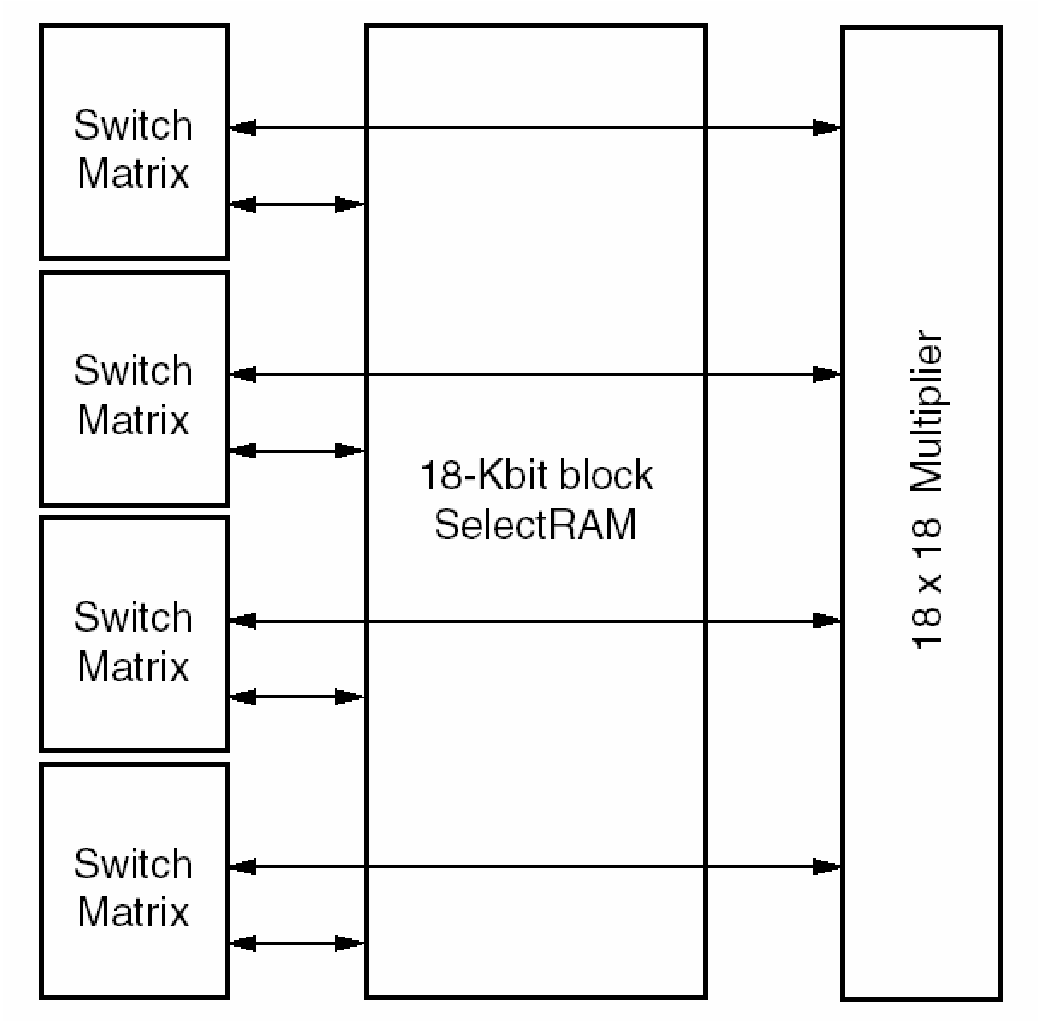


Embedded RAM

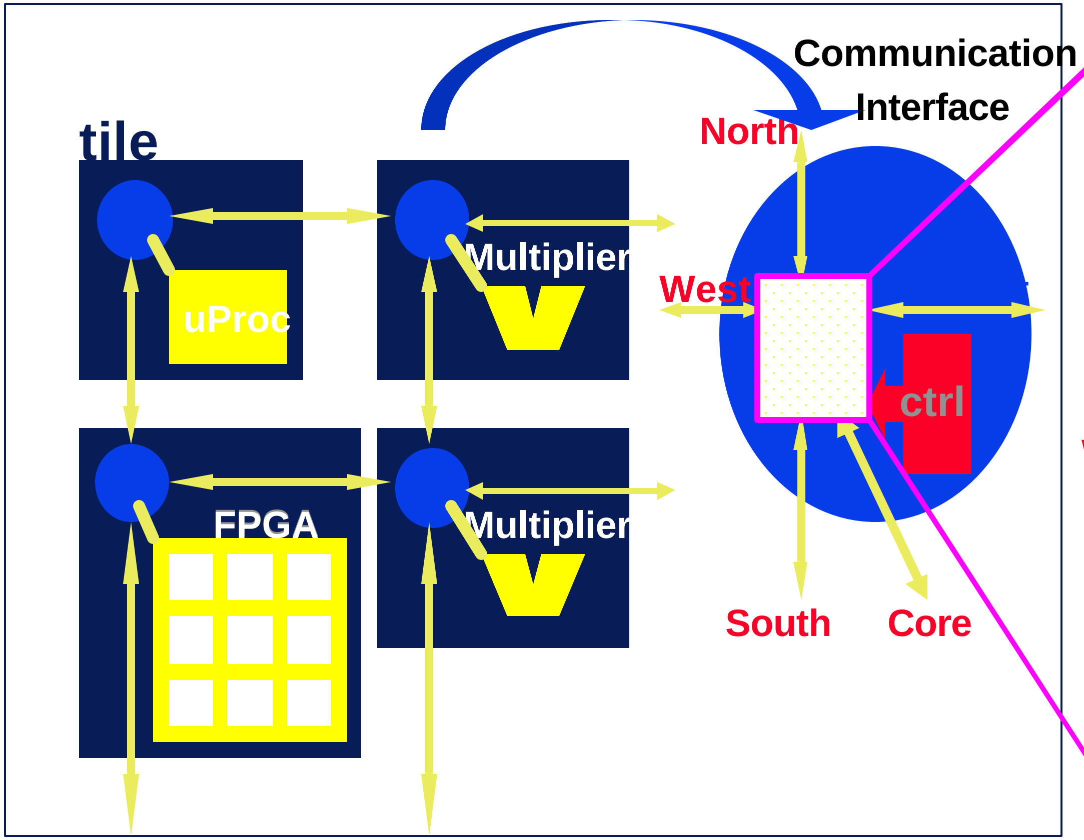
- **Xilinx – Block SelectRAM**
  - **18Kb dual-port RAM arranged in columns**
- **Altera – TriMatrix Dual-Port RAM**
  - **M512 – 512 x 1**
  - **M4K – 4096 x 1**
  - **M-RAM – 64K x 8**

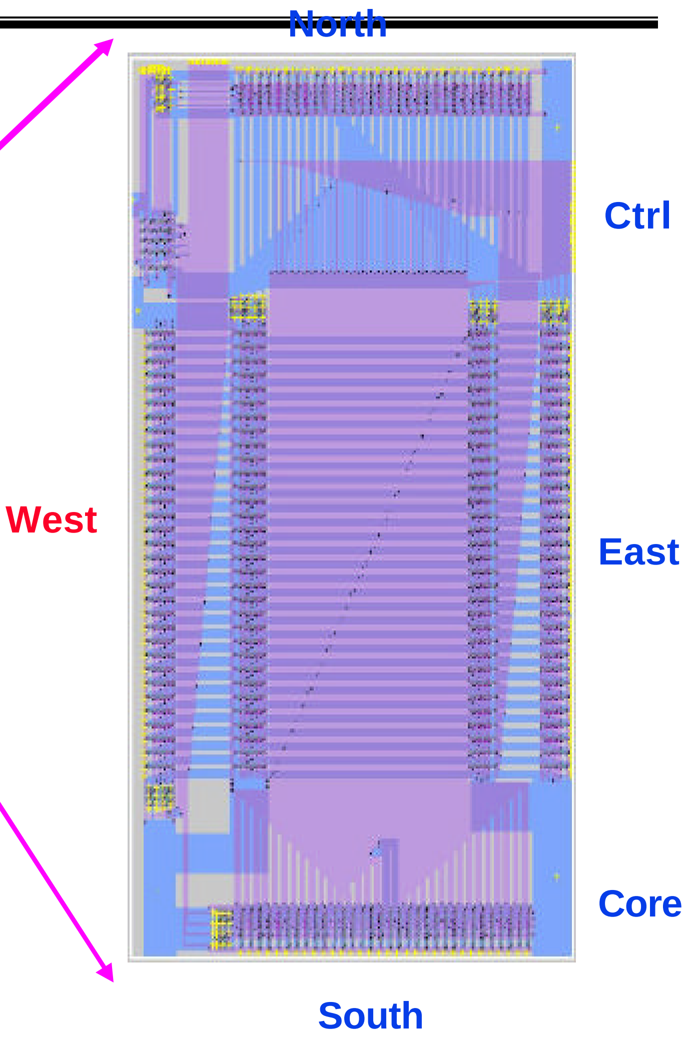


Xilinx: Embedded Multipliers



aSoC Architecture





- **Heterogeneous Cores**
- **Point-to-point connections**
- Communication Interface

Summary

- Reconfigurable computing relies heavily on new VLSI technology
- Device architectures maturing
- Application development progressing at rapid pace
- Integration of hardware and software a difficult challenge
- Active area of research at UMass.