
ECE 697F

Reconfigurable Computing

Lecture 7

Coarse Grained FPGA Architecture



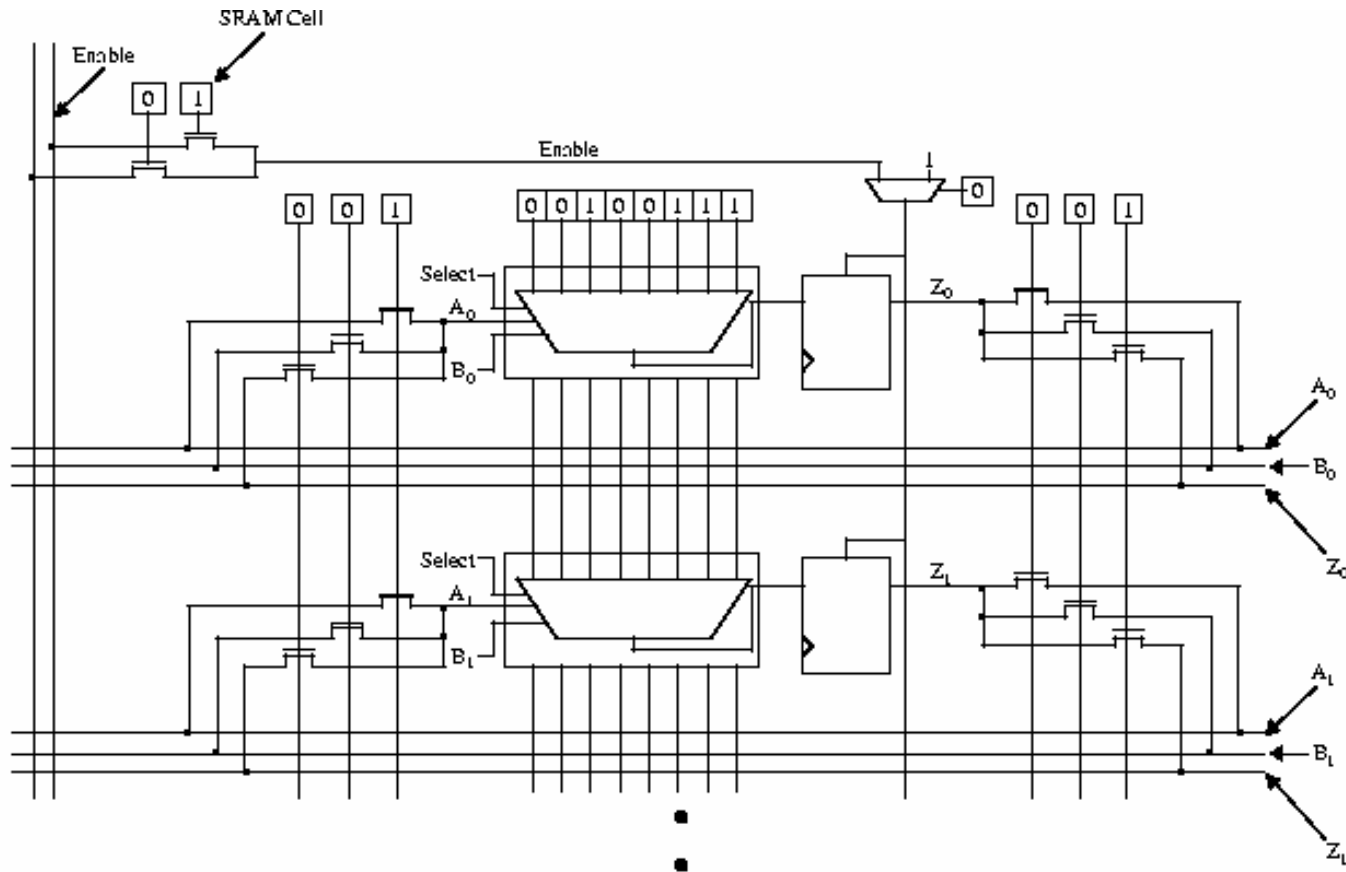
Overview

- **Review 5 different coarse-grained architectures**
- **How do they vary?**
 - **Basic block**
 - **Inter-block communication**
 - **Communication protocol**
 - **Software summary**
 - **Results**
- **Device trends**

Coarse-grained Architectures

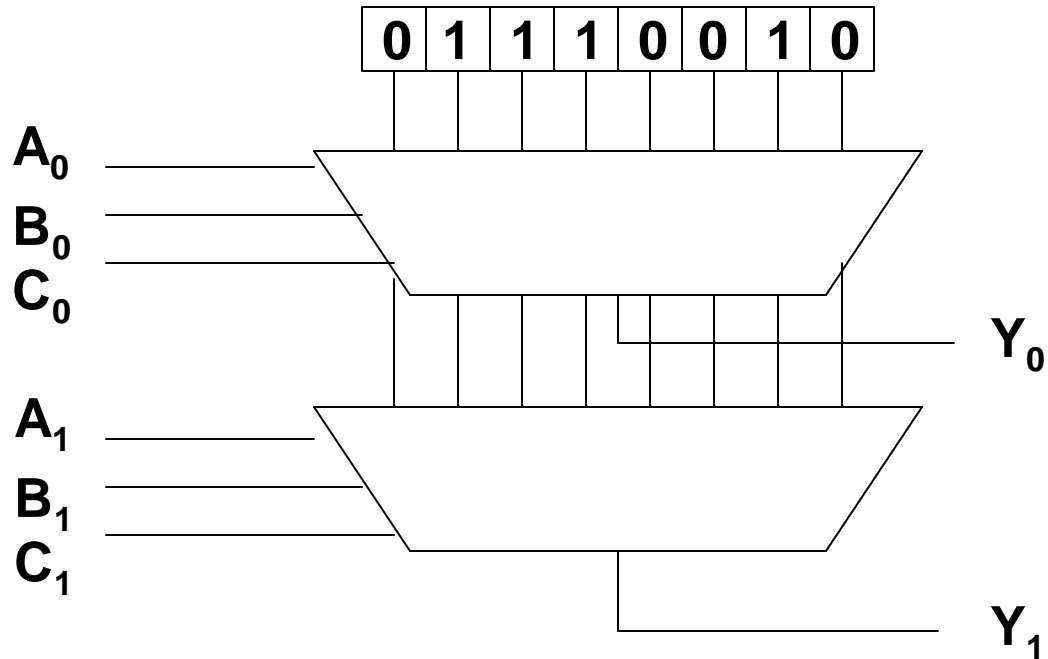
- **DP-FPGA**
 - LUT-based
 - LUTs share configuration bits
- **Rapid**
 - Specialized ALUs, multipliers
 - 1D pipeline
- **Matrix**
 - 2-D array of ALUs
- **Chess**
 - Augmented, pipelined matrix
- **Raw**
 - Full RISC core as basic block
 - Static scheduling used for communication

DP-FPGA



- Break FPGA into datapath and control sections
- Save storage for LUTs and connection transistors
- Key issue is grain size
- Cherepacha/Lewis – U. Toronto

Configuration Sharing



MC = LUT SRAM bits

CE = connection block pass transistors

$$A(N) = \frac{MC + N * CE}{N} = \frac{MC}{N} + CE$$

Set MC = 2-3CE

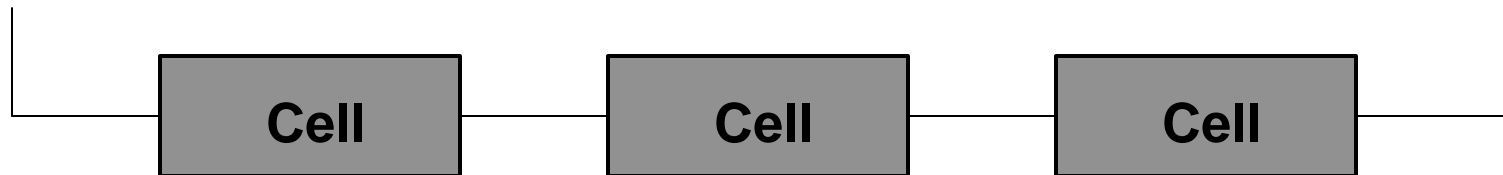
DP-FPGA Technology Mapping

- Ideal case would be if all datapath divisible by 4, no “irregularities”
- Area improvement includes logic values only.
- Shift logic included

| Circuit | A # Blocks Without Sharing | B # Blocks With Sharing | $\frac{A}{B}$ | Increase over Ideal | $\frac{\text{Area}(A)}{\text{Area}(B)}$ |
|---------|------------------------------------|---------------------------------|---------------|------------------------|---|
| awsim | 287 | 92 | 3.1 | 28% | 2.0 |
| coder | 673 | 193 | 3.5 | 14% | 2.2 |
| csamult | 516 | 162 | 3.2 | 26% | 2.0 |
| filter | 396 | 100 | 4.0 | 1% | 2.5 |
| fpadder | 395 | 123 | 3.2 | 24% | 2.0 |
| hubnet | 199 | 53 | 3.8 | 6% | 2.4 |
| mult | 64 | 16 | 4.0 | 0% | 2.5 |
| viterbi | 310 | 82 | 3.8 | 5% | 2.4 |

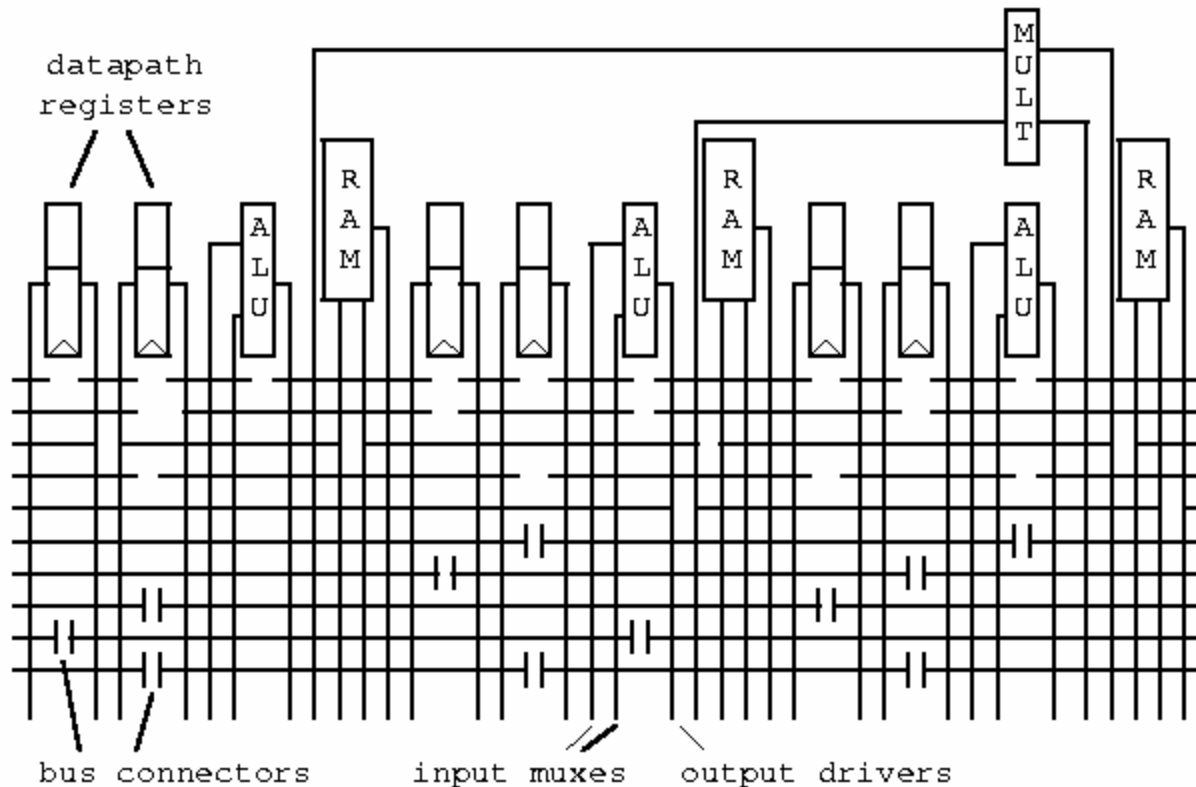
Rapid

- Reconfigurable Pipeline Datapath
- Ebeling –University of Washington
- Uses hard-coded functional units (ALU, Memory, multiply)
- Good for signal processing
- Linear array of processing elements.

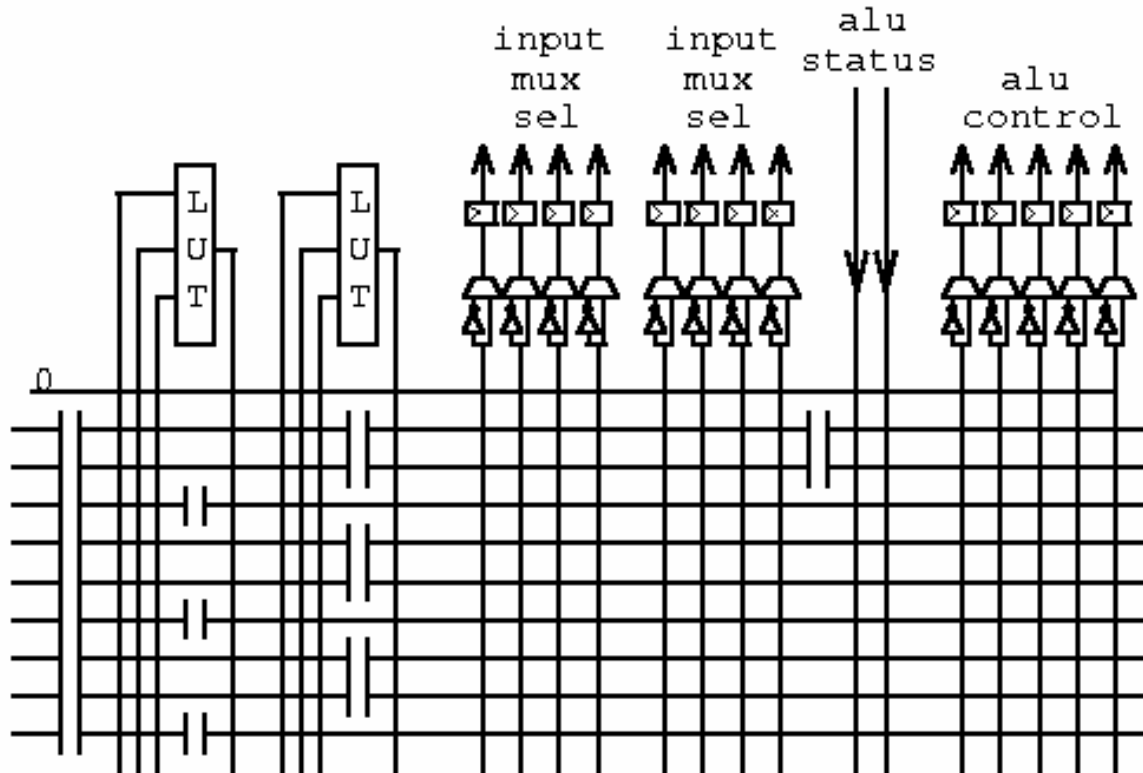


Rapid Datapath

- Segmented linear architecture
- All RAMs and ALUs are pipelined
- Bus connectors also contain registers



Rapid Control Path

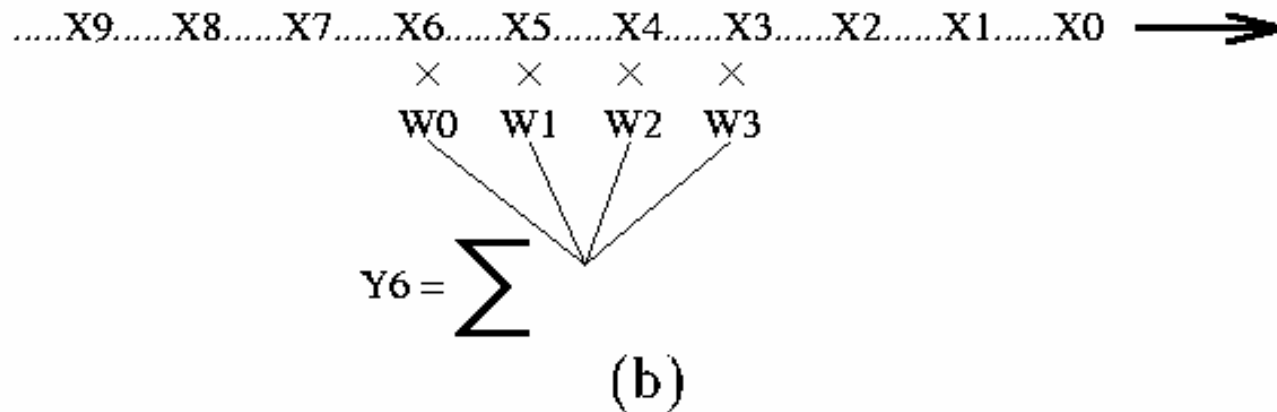


- In addition to static control, control pipeline allows dynamic control.
- LUTs provide simple programmability.
- Cells can be chained together to form continuous pipe.

FIR Filter Example

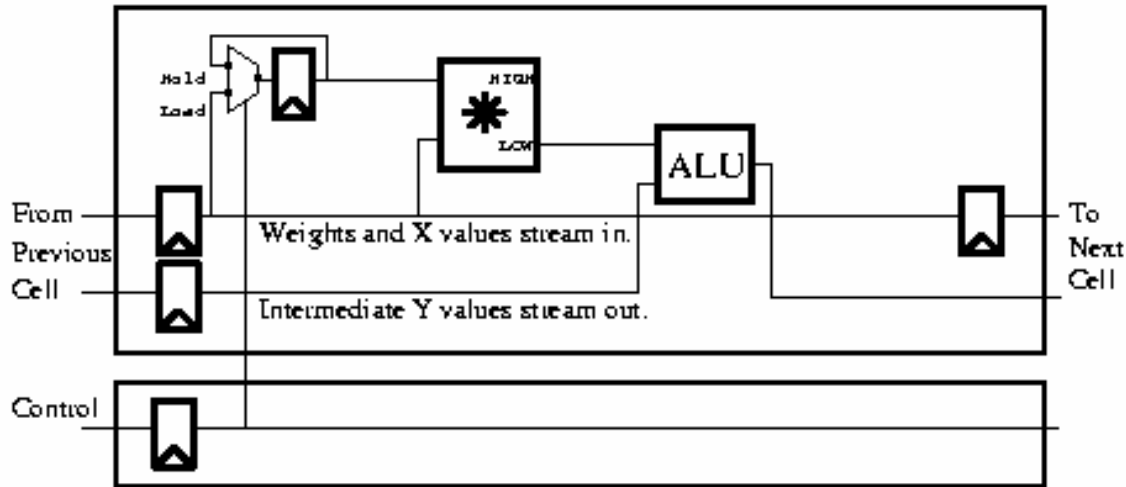
```
for i := NumTaps-1 to NumX-1
  Y[i] := 0
  for j := 0 to NumTaps-1
    Y[i] := Y[i] + X[i-j]*W[j]
  end
end
```

(a)

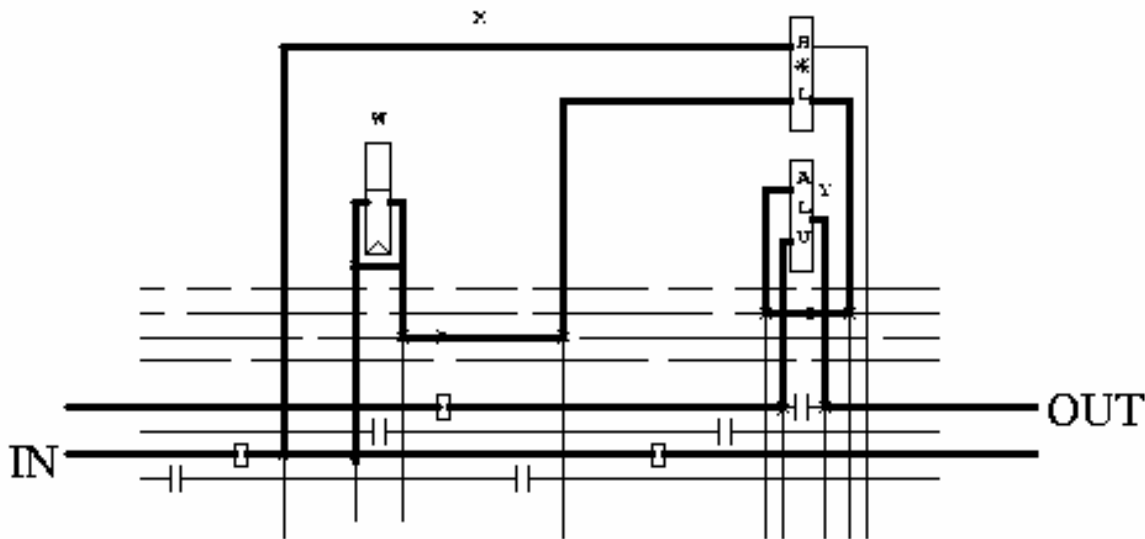


- Measure system response to input impulse
- Coefficients used to scale input.
- Running sum determined total.

Mapping a Tap to Rapid



(a)

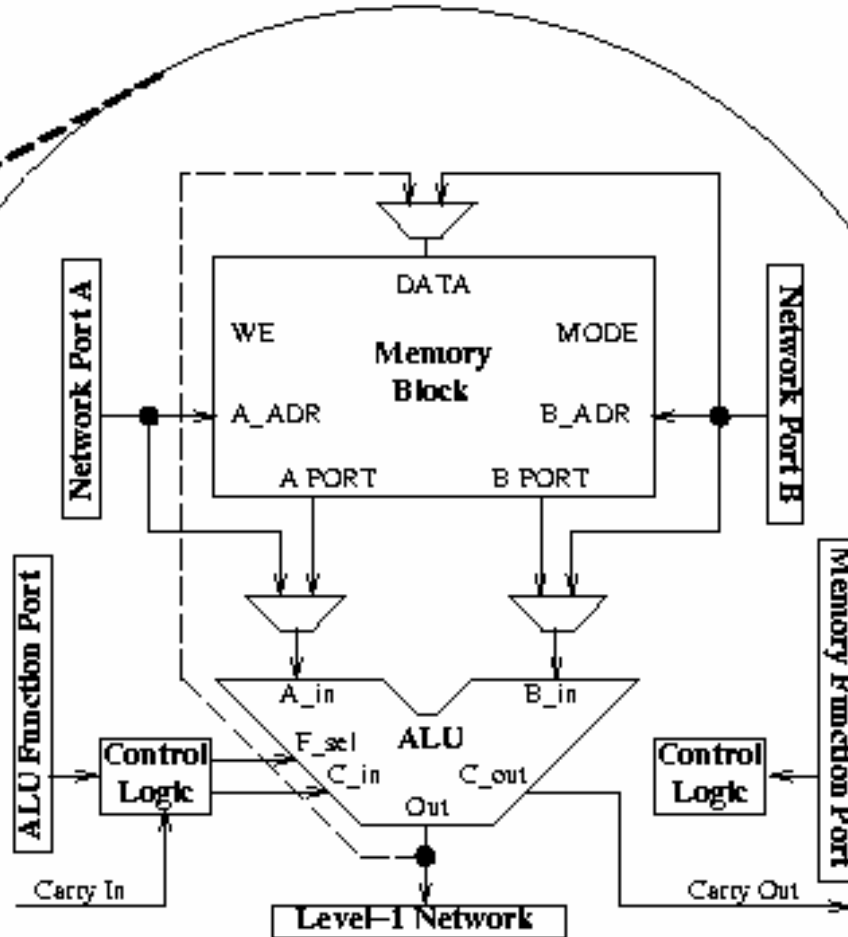


- Chain multiple taps together (one multiplier per tap)

Matrix

- Dehon and Mirsky -> MIT
- 2-dimensional array of ALUs
- Each Basic Functional Unit contains “processor” (ALU + SRAM)
- Ideal for systolic and VLIW computation.
- 8-bit computation
- Forerunner of SiliconSpice product.

Basic Functional Unit

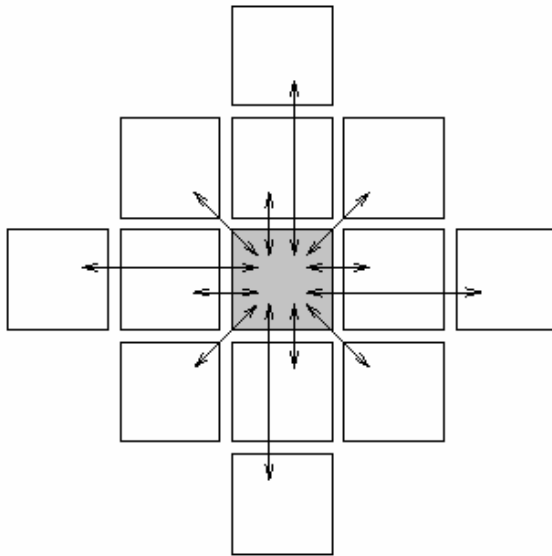


- Two inputs from adjacent blocks.
- Local memory for instructions, data.

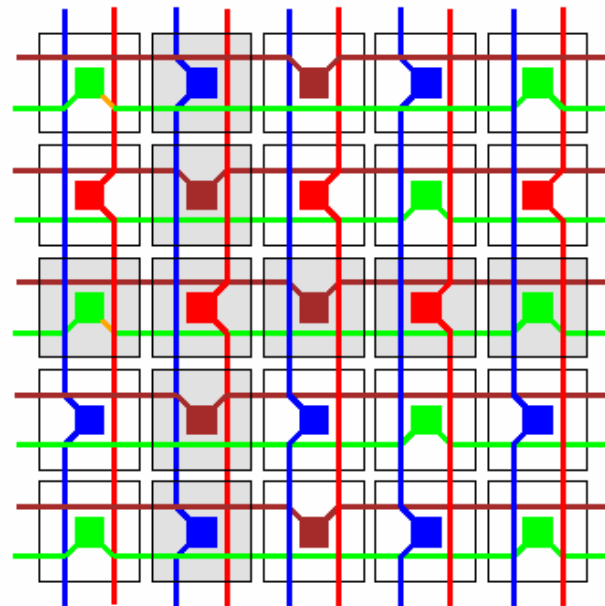
BFU Interconnect

- Near-neighbor and quad connectivity
- Pipelined interconnect at ALU inputs
- Data transferred in 8-bit groups.
- Interconnect not pipelined

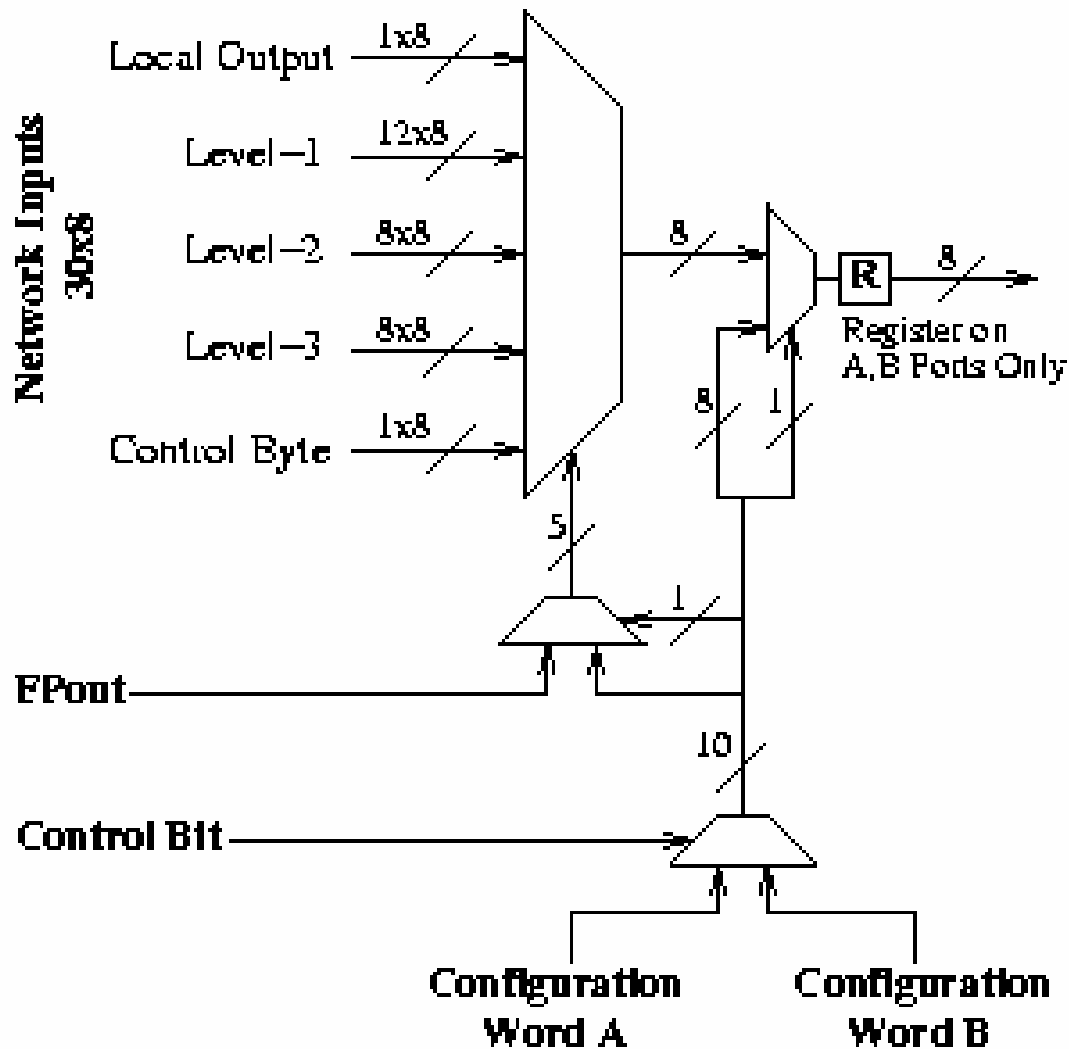
Nearest Neighbor Interconnect



Length Four Bypass Interconnect



BFU Inputs



BFU (A,B)
Network Drivers (N1,N2)

- Each ALU inputs come from several sources.
- Note that source is locally configurable based on data values.

Filtering Example

- $y_i = w_1 * x_i + w_2 * x_{i+1} + w_3 * x_{i+2} + \dots + w_k * x_{i+k-1}$
- For k-weight filter 4K cells needed
 - One result every 2 cycles
- K/2 8x8 multiplies per cycle

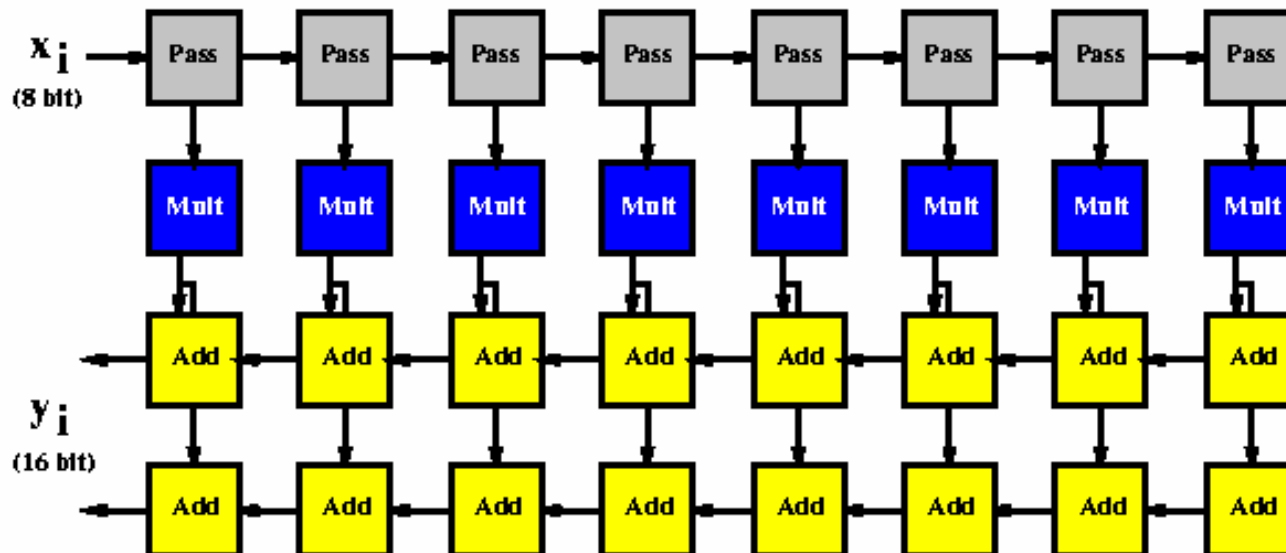
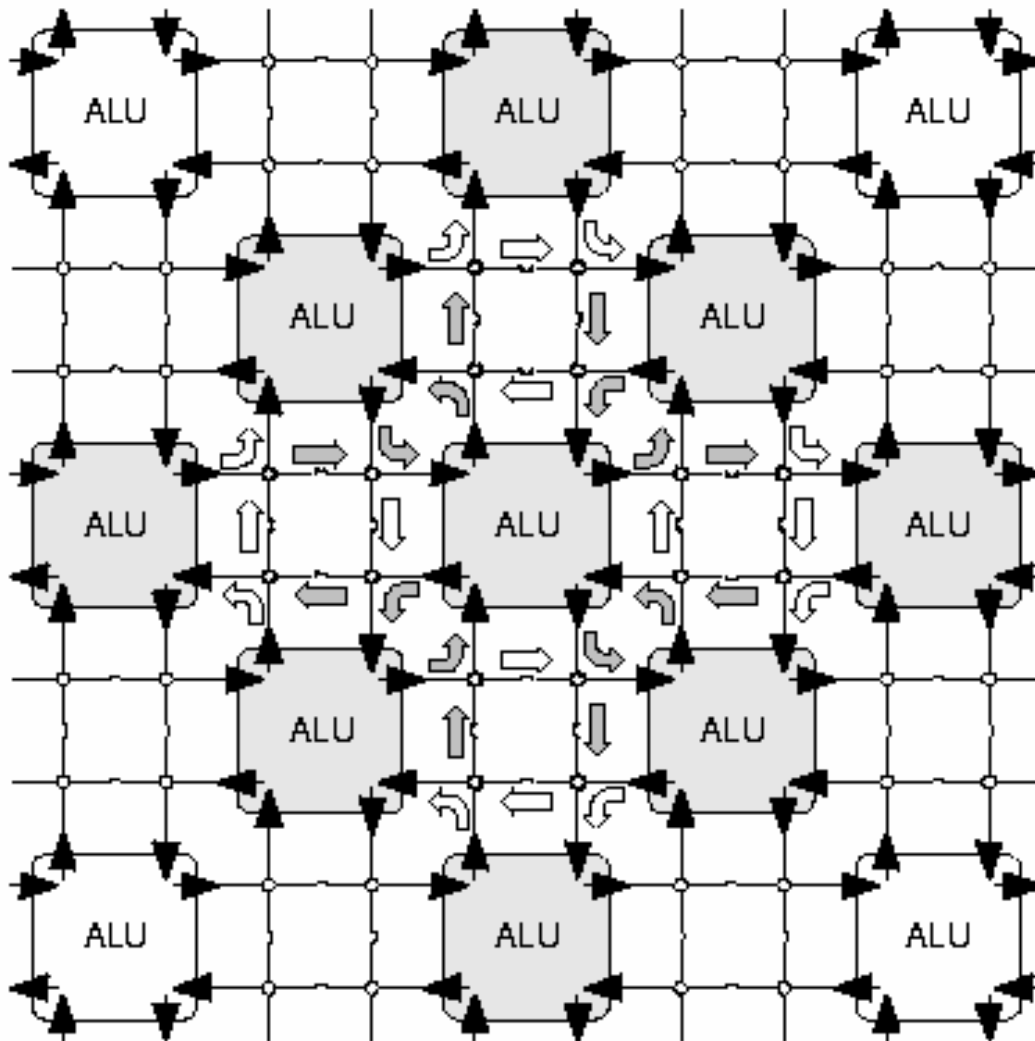


Figure 5: Systolic Convolution Implementation

Chess

- **HP Labs – Bristol, England**
- **2-D array – similar to Matrix**
- **Contains more “FPGA-like” routing resources.**
- **No reported software or application results**
- **Doesn’t support incremental compilation**

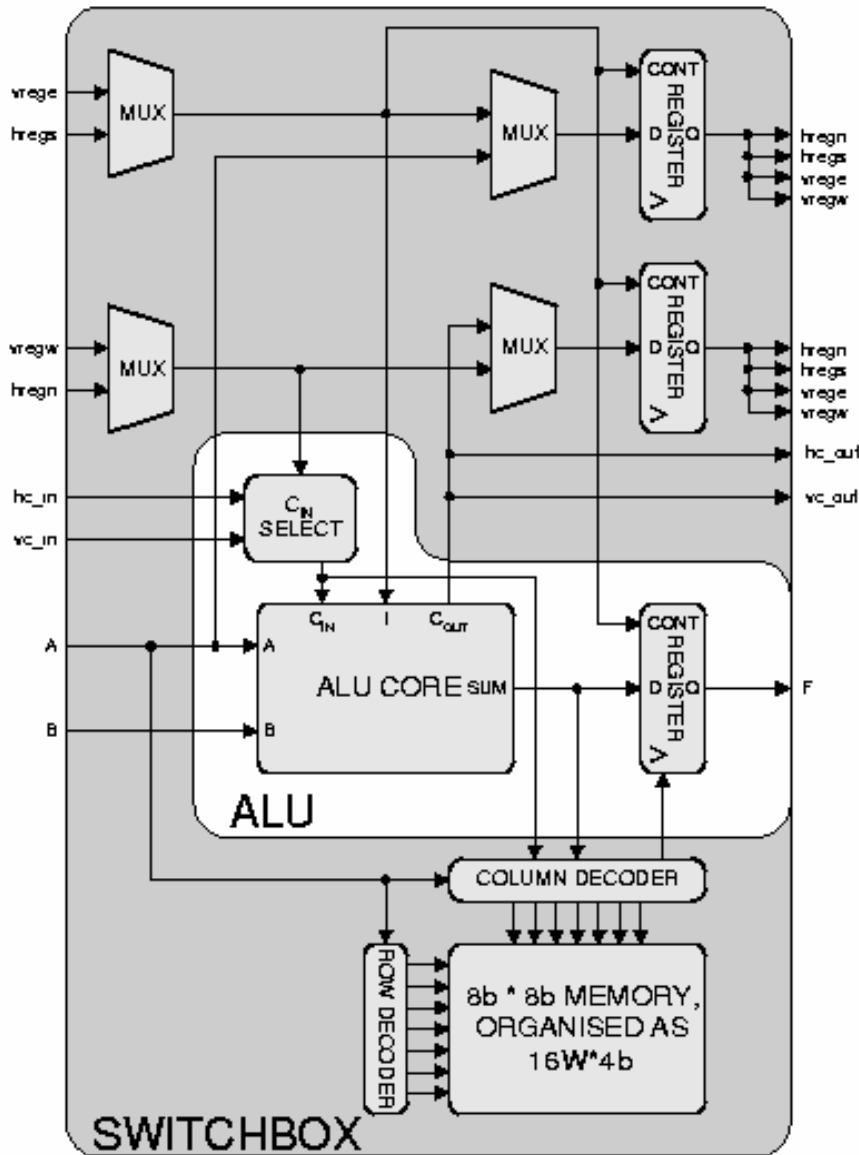
Chess Interconnect



- More like an FPGA
- Takes advantage of near-neighbor connectivity

Figure 1 CHES layout and nearest neighbour wiring

Chess Basic Block



- Switchbox memory can be used as storage
- ALU core for computation

Figure 4 Logical structure of an ALU and switchbox

Chess Statistics

- Use metrics to evaluate computational power.
- Efficient multiplies due to embedded ALU
- Process independent.

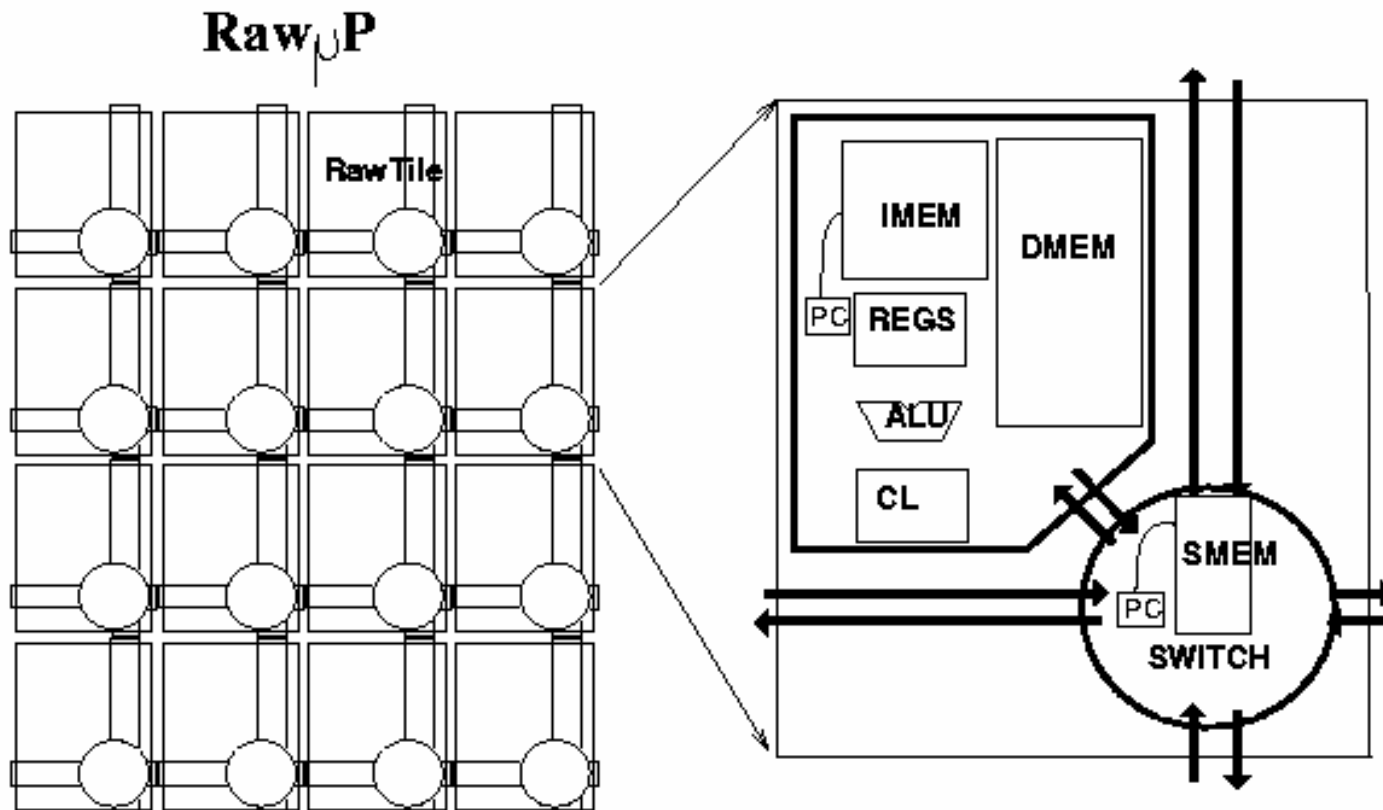
| | X3K (CLB) | X4K (CLB) | X6K (CELL) | CHESS (ALU + Sbox) |
|---------------------------|--------------|--------------|---------------|-----------------------|
| Process (μ) | 1.2 | 1.2 | 0.6 | 0.35 |
| λ (μ) | 0.6 | 0.6 | 0.3 | 0.18 |
| Area (mm^2) | 0.47 | 0.45 | 0.020 | 0.043 |
| Area/M λ^2 | 1.30 | 1.25 | 0,22 | 1.41 |
| ops/cell/cycle: | | | | |
| bit ops | 2 | 32 | 1 | 68 |
| lut4 ops | 2 | 2 | 0.17 | 4 |
| alu ops | 1 | 2 | 0.33 | 4 |
| mul ops | 0.67 | 0.67 | 0.25 | 4 |
| ops/M λ^2 /cycle: | | | | |
| bit ops | 1.5 | 25 | 4.6 | 48 |
| lut4 ops | 1.5 | 1.6 | 0.8 | 2.8 |
| alu ops | 0.8 | 1.6 | 1.5 | 2.8 |
| mul ops | 0.5 | 0.5 | 1.1 | 2.8 |

Reconfigurable Architecture Workstation

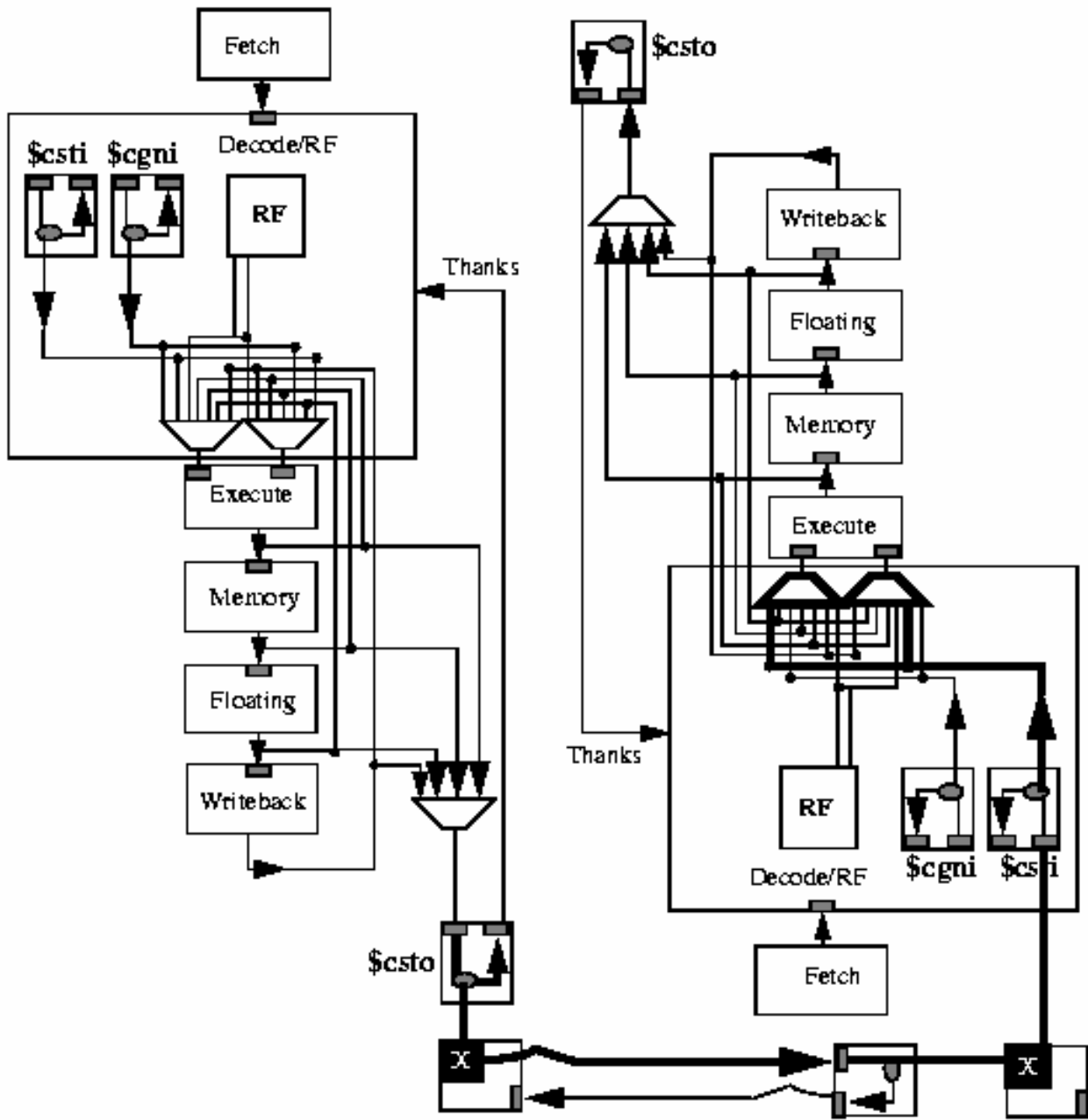
- **MIT Computer Architecture Group**
- **Full RISC processor located as processing element.**
- **Routing decoupled into switch mode**
- **Parallelizing compiler used to distribute work load.**
- **Large amount of memory per tile.**

RAW Tile

- Full functionality in each tile
- Static router located for near-neighbor communication



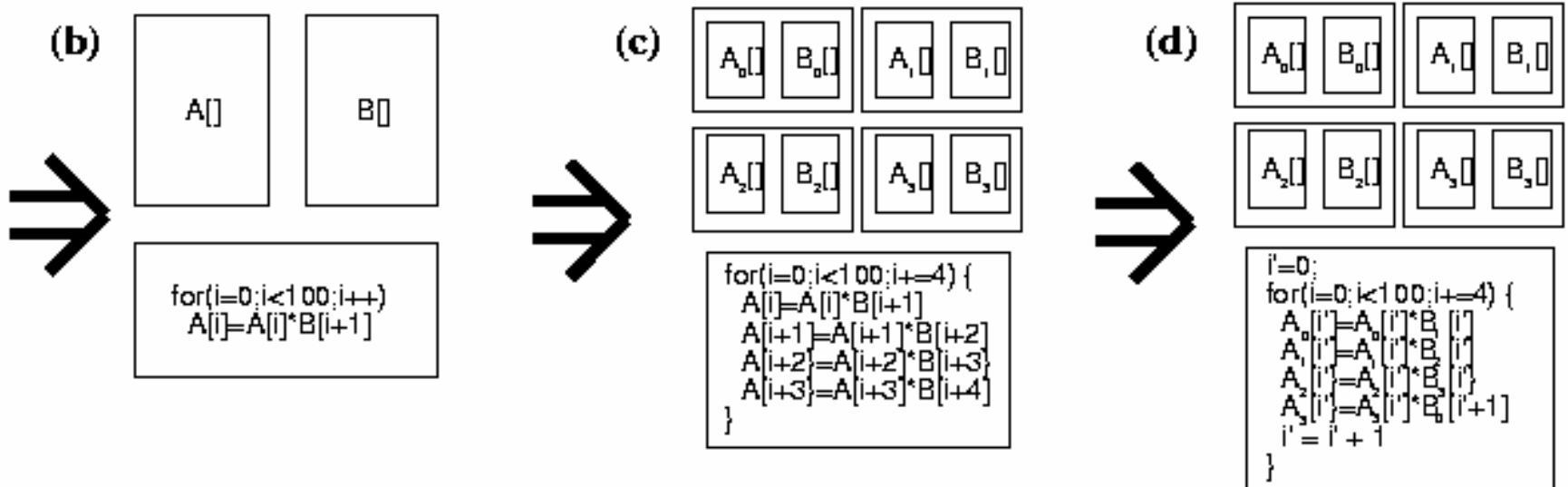
RAW Datapath



The Processor-Switch-Switch-Processor path

Raw Compiler

- Parallelizes compilation across multiple tiles
- Orchestrates communication between tiles
- Some dynamic (data dependent) routing possible.



Summary

- **Architectures moving in the direction of coarse-grained blocks.**
- **Latest trend is functional pipeline.**
- **Communication determined at compile time**
- **Software support still a major issue.**