CS294-7: Reconfigurable Computing

Day 5: Tuesday, February 4

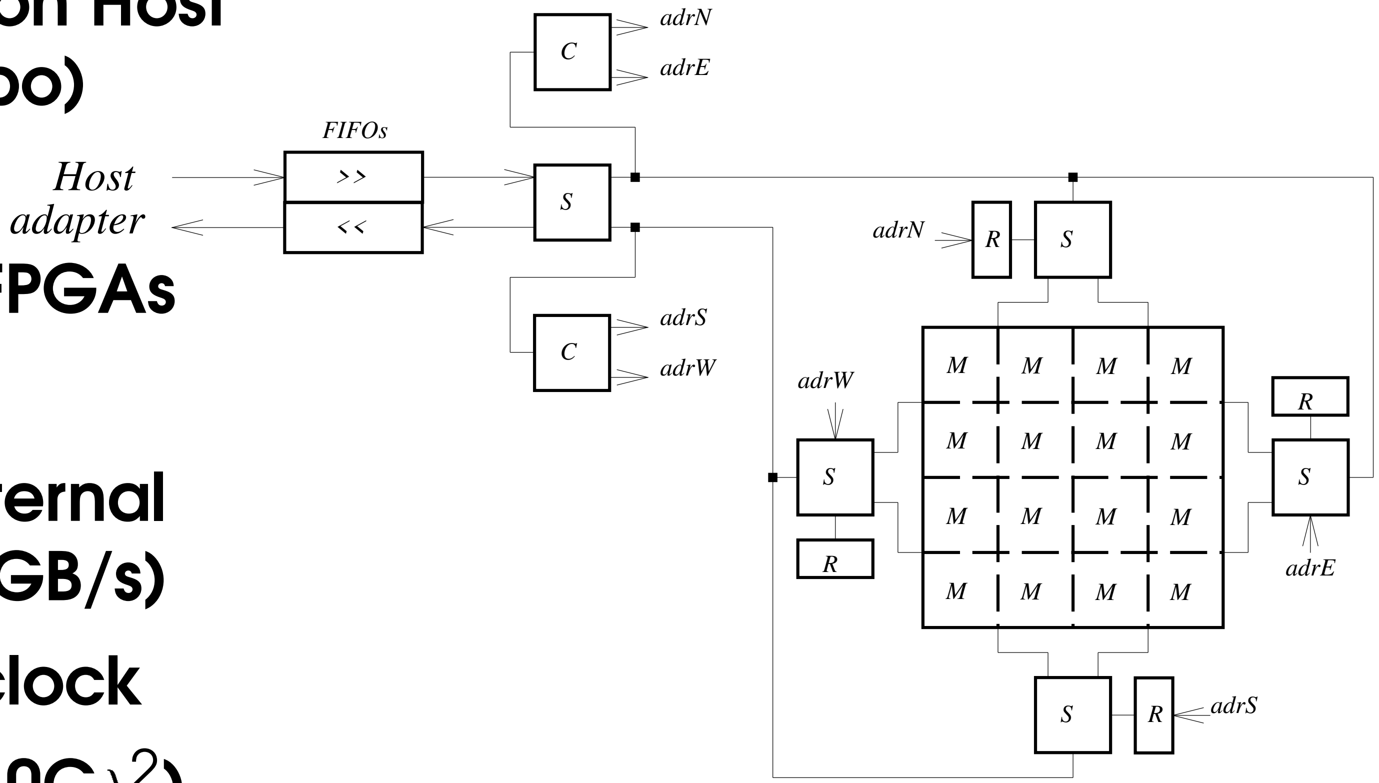
- Admin
- Programmable Active Memories
- Cheops

Admin

- **Assignment 1 Handout**
  - **Proposals:** February 13 **(9 days)**
  - **Due:** March 6
  - **Tools handout Thursday**
- **Viewlogic poll**
- **This Friday, 2 pm, RC talk at ICSI**

PAM Architecture

- Workstation Host (DEC, turbo)
- FIFO trns.
- Array of FPGAs
- Memory
- direct external data (1.2GB/s)
- tunable clock
- Perle-1 (30Gλ^2)
  - 23 XC3090s (420Mλ^2) ($\approx$ 15K 4-LUTs)
  - 4MB SRAMs (20Gλ^2)



PAM Programming

- synthesis too wasteful?
- schematic capture too painful?

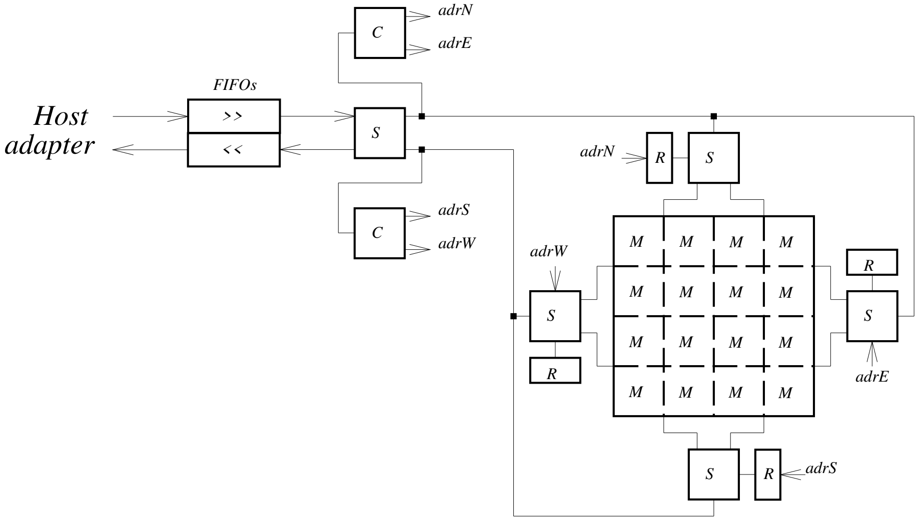
- Algorithmic description of structure
  - netlist generator

Claim: PAM easier than ASIC development

```
void logic(Net<N>& a, Net<N>& b, Net<N>& c,
           Net<N>& sum, Net<N>& carry) {
  input(a); input(b); input(c);
  output(sum); output(carry);
  for (int i=0; i<N ; i++) {
    sum[i] = a[i]^b[i]^c[i];
    carry[i] = (a[i] & b[i])
             | (b[i] & c[i])
             | (c[i] & a[i]);
    }
  }
void placement(Net<N>& sum, Net<N>& carry) {
  for (int i=0; i<N ; i++) {
    carry[i] <<= sum[i];
    sum[i+1] <<= sum[i] + OFFSET(0,1);
    }
  }
```

Interfacing

- Unix device driver
- FIFO datastreams to/from host
- TURBO bus (100MB/s ?)
- Long latency from processor (μs)



Applications

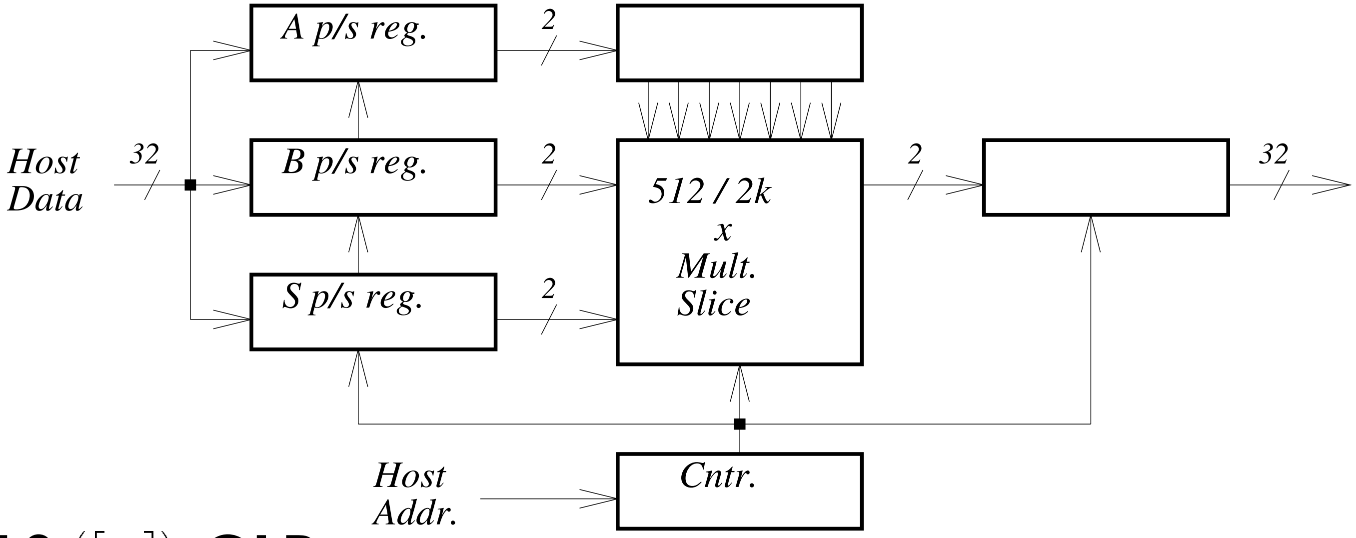
1. Bignum Multiply (faster than Cray II, Cyber)
2. RSA (faster than custom)
3. Edit distance (SPLASH next time)
4. Wave equations (faster than supercomputers)
5. Neural net
6. Video compression (JPEG, MPEG)
7. Image acquisition
8. Image feature extraction (faster than all alternatives)
9. Stereo vision (30-200$\times$ faster than μP/DSPs)
10. Sound synthesis (256 voices, 10$\times$ DSP capacity)
11. Viterbi decode

Bignum Multiply

- long-word $\times$ 2-bit multiply
- 33MHz

334 bits

LSA	2μs
PAM	5μs
Cray	55μs

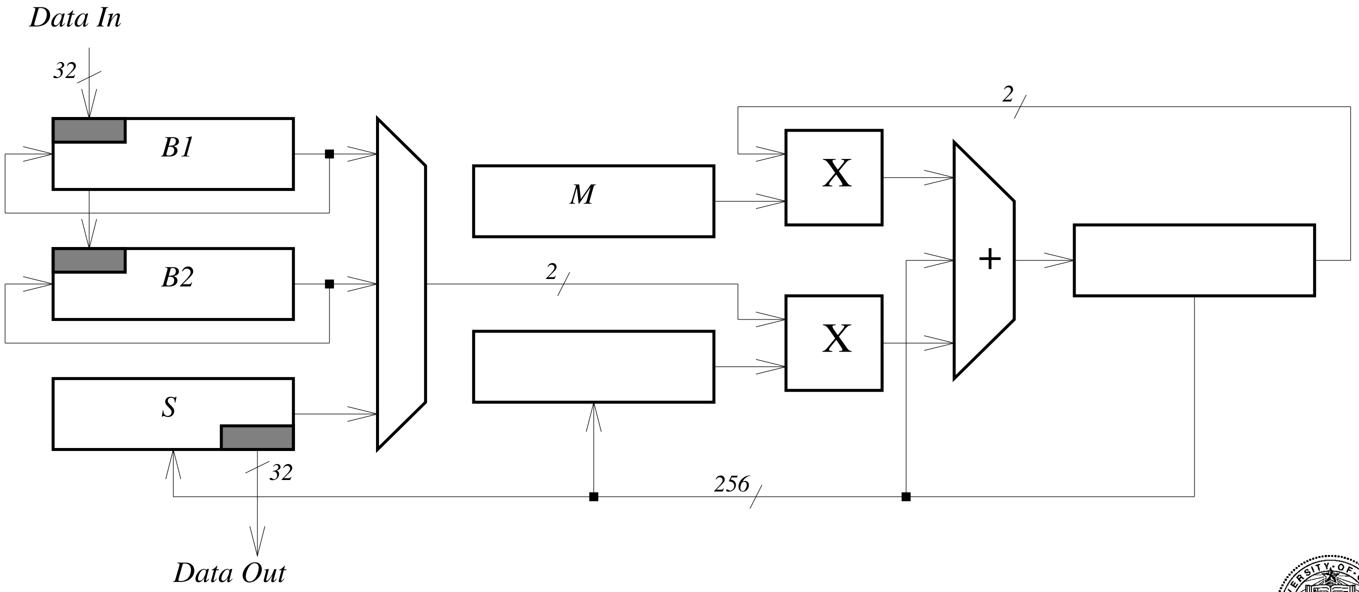


LSA

Area	$30 + 18\left(\left\lceil\frac{n}{4}\right\rceil\right)$ CLBs
Time	$\left(4 + \left\lceil\frac{n}{2}\right\rceil\right) 12.7$ ns

RSA

- $\approx 1K$ **key at 185Kb/s**
- ≈ 512 **key at 300Kb/s**
- **1992 Alpha 512b at 56Kb/s (21064 – 150MHz – 1.7Gλ^2, $\lambda = 0.375\mu$)**



RSA

- specialized around modulus
- Montgomery modulus
- exploit problem structure
- heavy pipelining

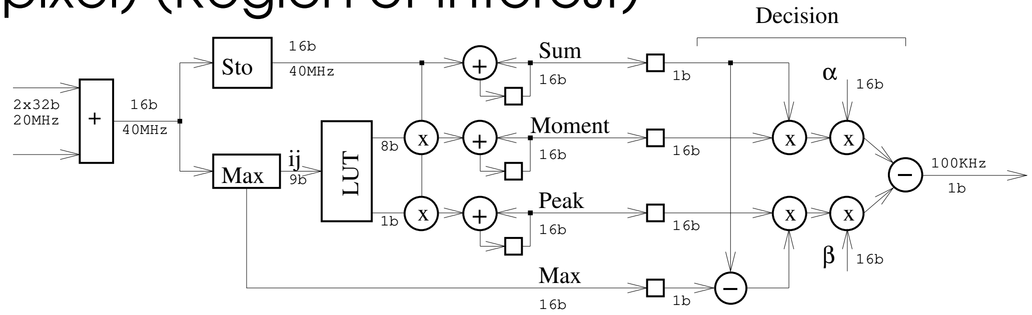
Chinese Remainder Theorem	4
Precompute powers	1.25
Hensel's division	1.5
log-time adder	≈ 2
Quotient pipelining	4

Sensor Processing

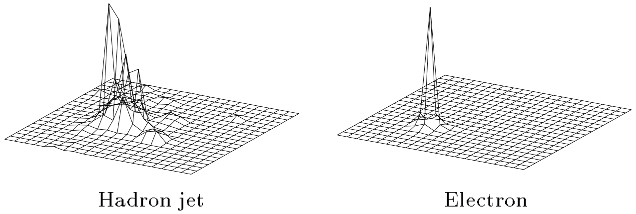
- Continuous input
- Large volume of data input (GB/s)
- Preprocessing/reduction
- Examples
  - SONAR/RADAR
  - High Energy Physics

Calorimeter

- 100kHz real-time constraint
- 20×20 pixels (2 16b ints/pixel) (Region of Interest)
- 160MB/s input data



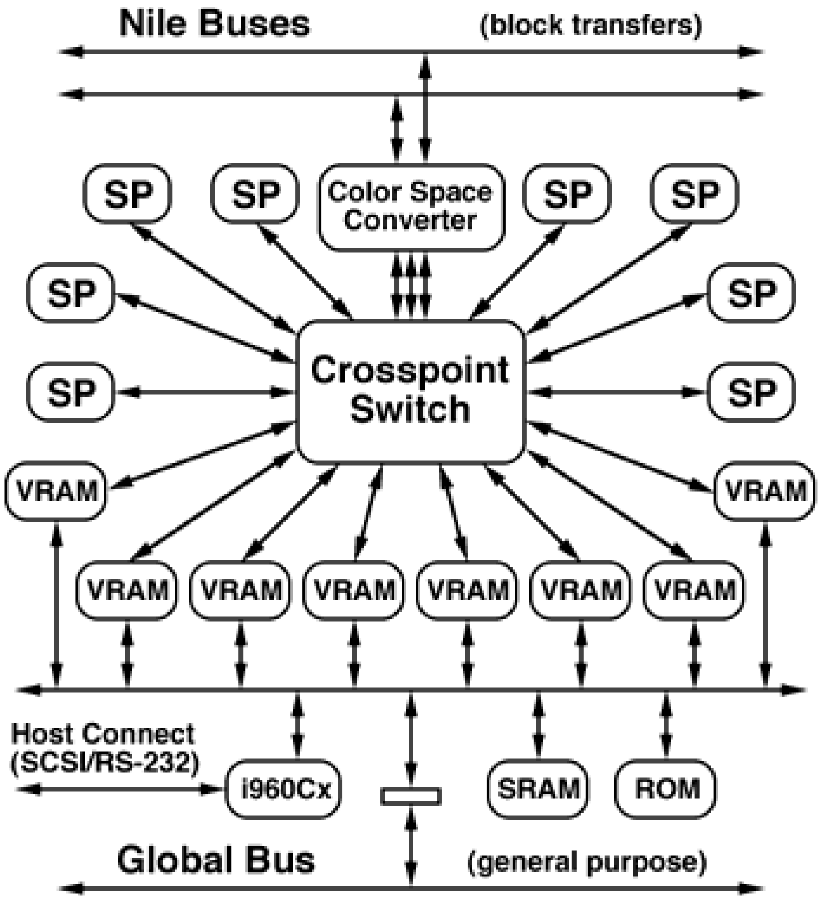
- Compute
  - pixel-wise sum
  - total energy
  - max energy
  - first moment
  - peak energy
  - discriminant



- 33G bit ops
- 40MHz PAM – meets 10μs target (one-quarter of Perle-1)
- 12K RISC instructions? (per ROI)
- 7K RISC instructions? (w/ hdw mpy)
- SPARC10 takes 1290μs (129× slower)

Cheops

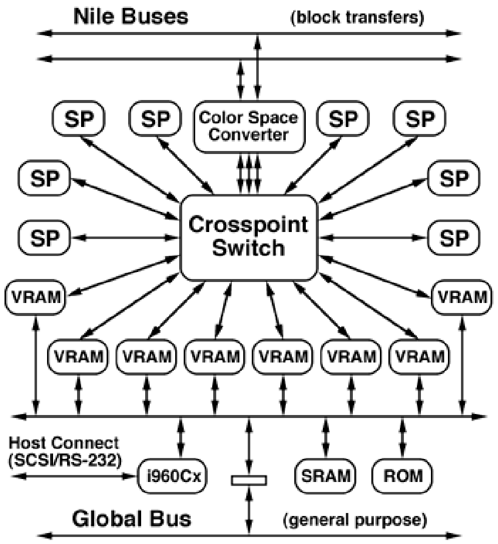
- dedicated compute elements
- monolithic crossbar
- "modular"
- scalable (?)

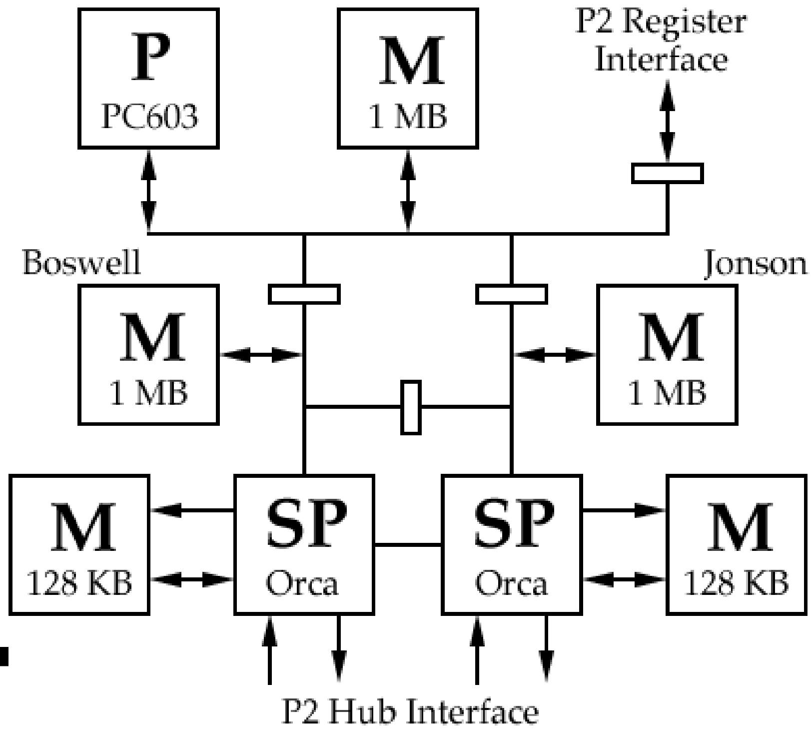


Target Domain: Video Processing

Stream Processors

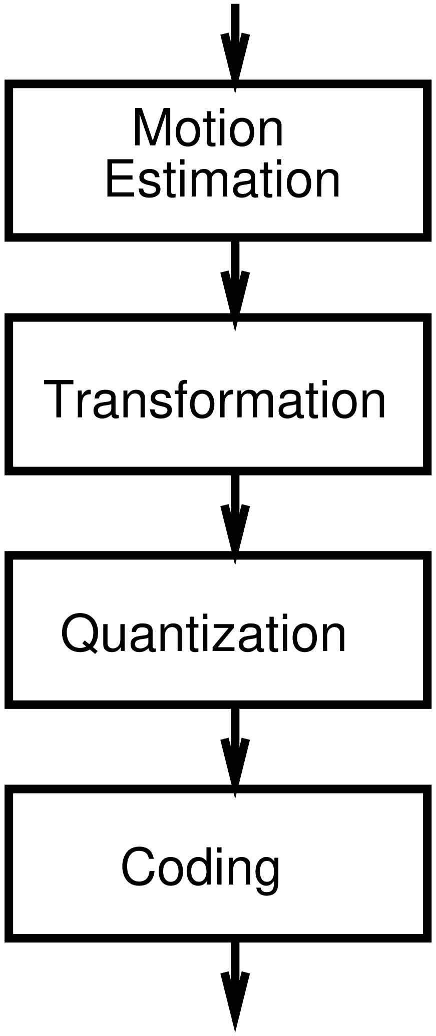
1. FIR / matrix multiply
2. MAC
3. DCT
4. Motion estimation
5. colorspace conversion
6. FPGA+SRAM
7. RISC processor



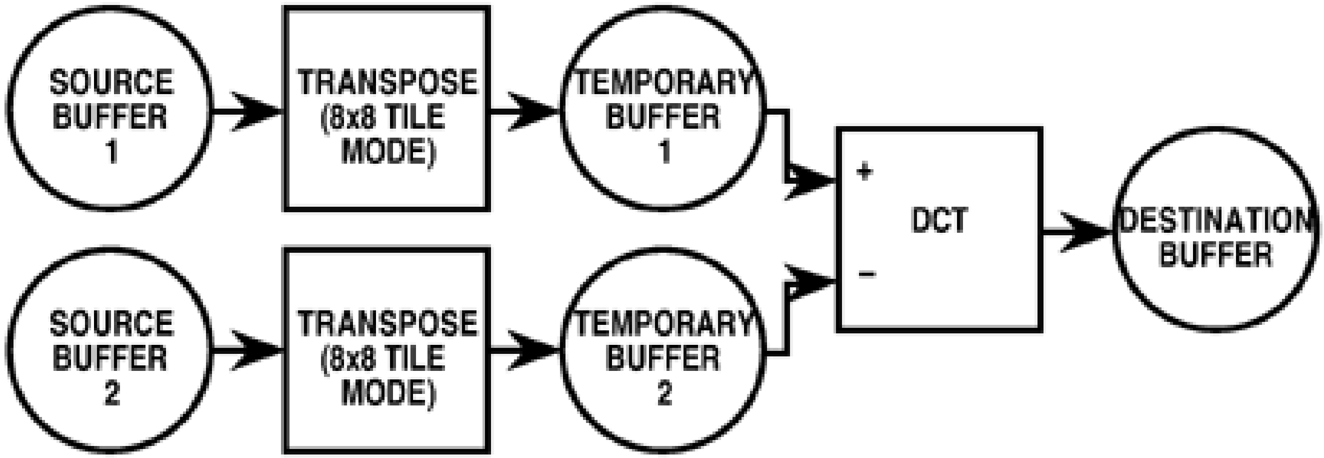


Programming

- **Stream Processing**
- **Pipelined dataflow between operators (*e.g.* SDF)**



Example



SOURCE BUFFER 1
TRANSPOSE (8x8 TILE MODE)
TEMPORARY BUFFER 1
SOURCE BUFFER 2
TRANSPOSE (8x8 TILE MODE)
TEMPORARY BUFFER 2
+
DCT
−
DESTINATION BUFFER

Review: Common Themes

Two CCMs:

- PAM (FPGA+memory)
- Cheops (coarse-grain blocks)

Both wanted:

- High Performance (often real-time constraints)
- Programmability

Both exploited:

- Spatial Processing
- on Streams of Data