CS294-7: Reconfigurable Computing

Day 23: Tuesday, April 15

Retiming

- Admin
- Retiming in the Large
- Retiming Structures
- Retiming Requirements

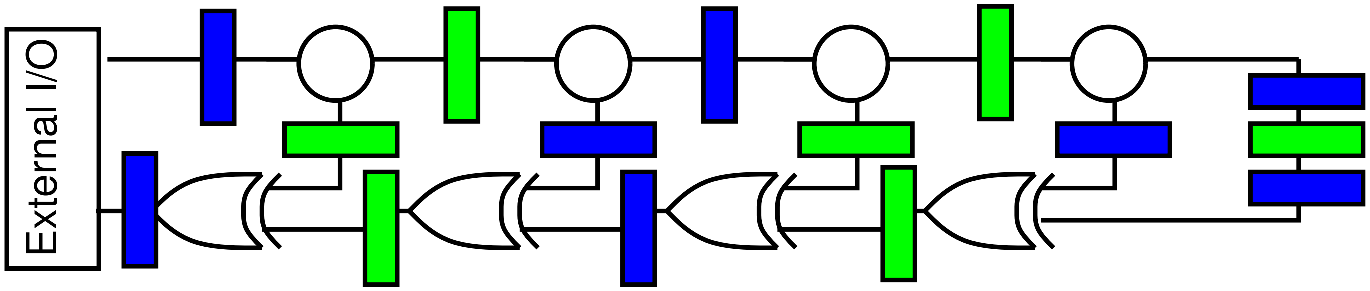
Admin

- No Class Thursday (4/17) $\rightarrow$ FCCM
- Projects Due in 3 weeks
- Reorder next four lectures
  - Multicontext
  - Control
  - Specialization
  - Programming Styles

Retiming

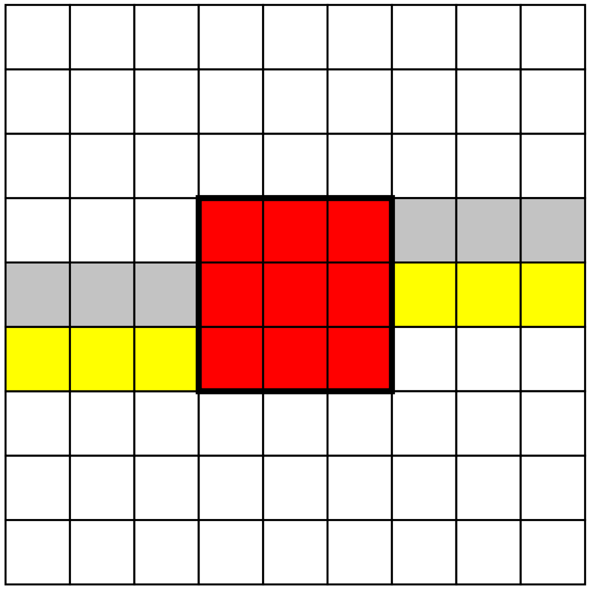
Last time (Day 22):

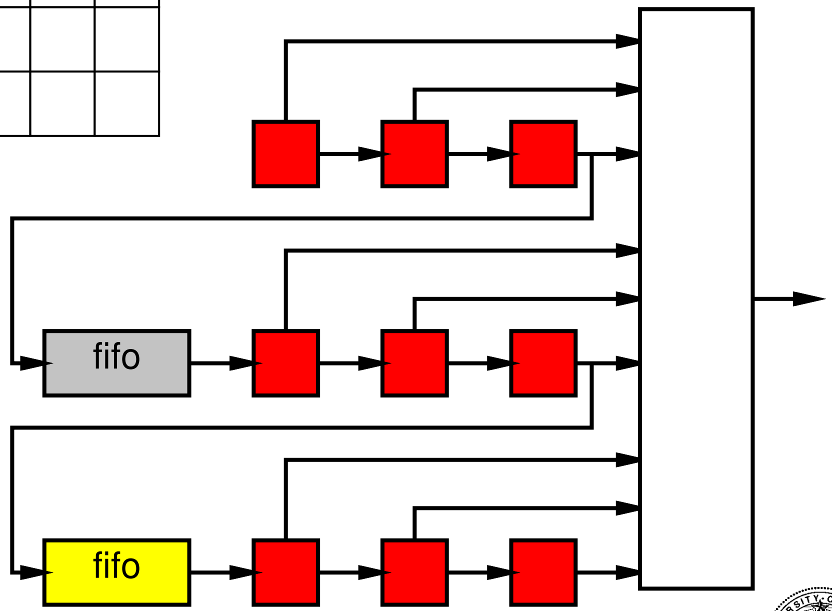
- Saw we could retime to reduce cycle time
- By accepting C-slow or multicontext implementations have retiming registers around every compute stage



Retiming in the Large

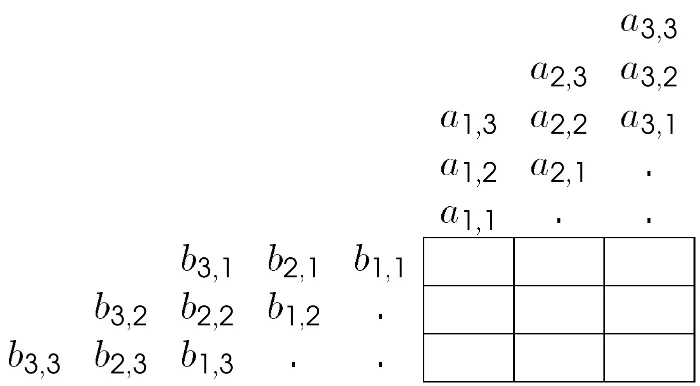
Think back to RSP:

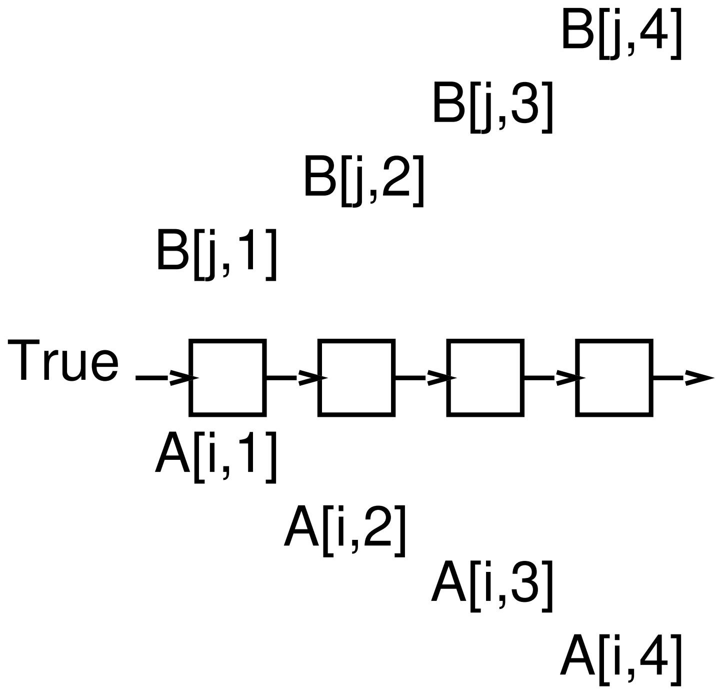




Retiming in the Large

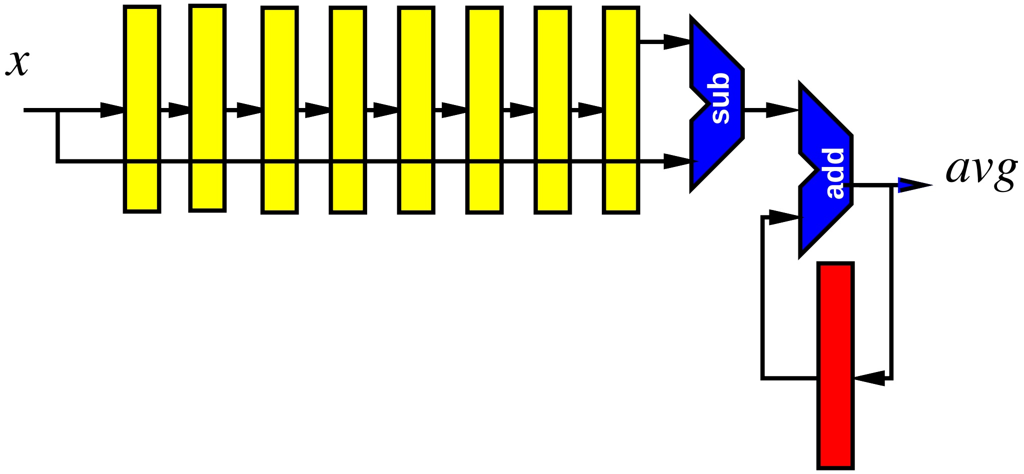
Systolic structures:





Retiming in the Large

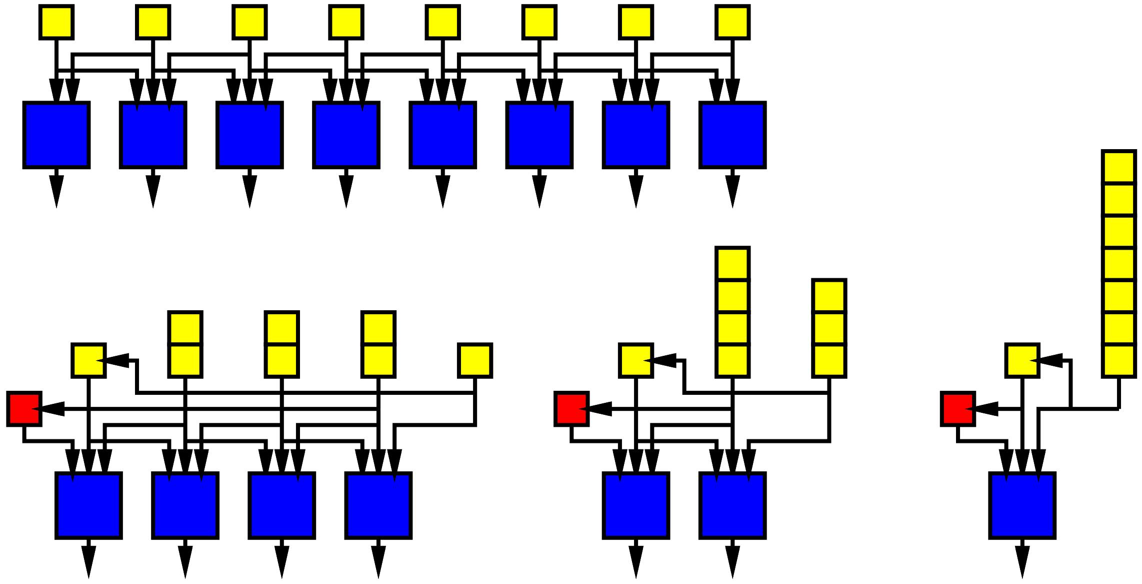
Averaging Filter:



Similarly: echo cancellation

Serialization → Retiming

In general, serialization means retime data
→ deeper retiming per compute.



Viewpoint

- Retiming $\equiv$ temporal interconnect
- Role of *data* memory to perform retiming

Implementation

How does programmable retiming get implemented?

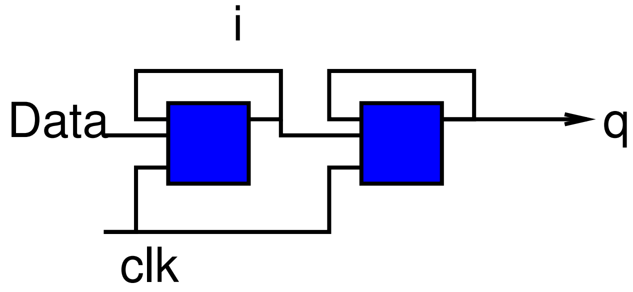
Concerns:

- **Area:** λ^2**/bit**
- **Time: bandwidth**

N.B. latency important when we do not know when we need data item again

Retiming Structures

Build flip-flop out of logic blocks:

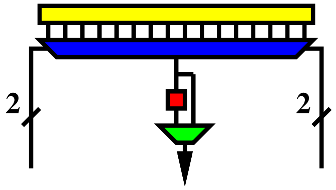


$$i \leftarrow Data \cdot \overline{clk} + i \cdot clk \qquad q \leftarrow q \cdot \overline{clk} + i \cdot clk$$

Area	**2 LUTs (800Kλ^2 ea.)**
Bandwidth	**1b/cycle**

Retiming Structures

Real flip-flop (optionally) on output:

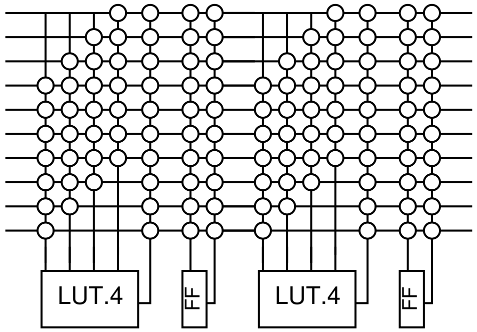


Flip-flop	4-5Kλ^2
Switch select	$\approx$5Kλ^2

Area	1 LUT (800Kλ^2 ea.)
Bandwidth	1b/cycle

Retiming Structures

Separate flip-flops:



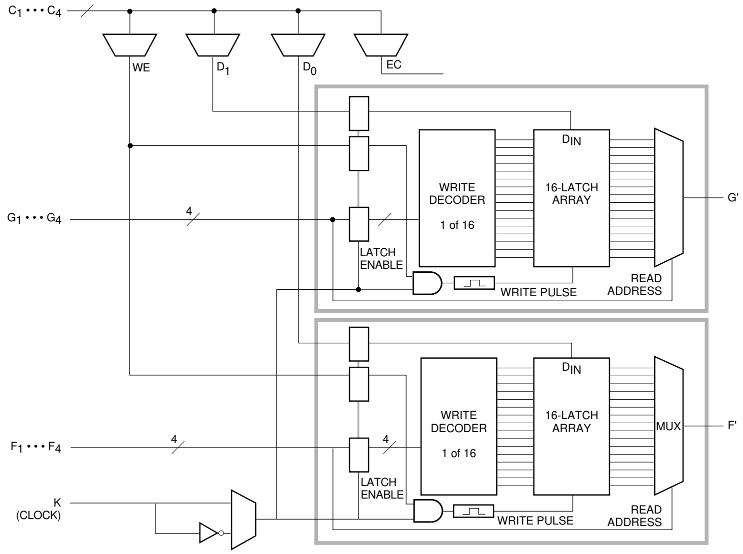
Assume Routing requirements go as inputs

$\rightarrow$ one-fourth LUT size

Area	(≈200Kλ^2 ea.)
Bandwidth	1b/cycle

Retiming Structures

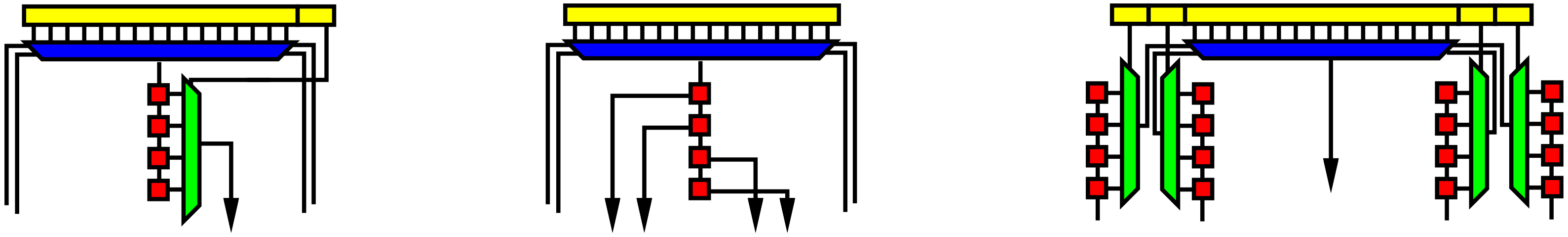
Xilinx CLB as memory:



Area	**$\approx$640Kλ^2/16 $\approx$40K λ^2/bit**
Bandwidth	**1b/cycle (5.5 ns/read; 8 ns/write)**
	1/16 th of capacity

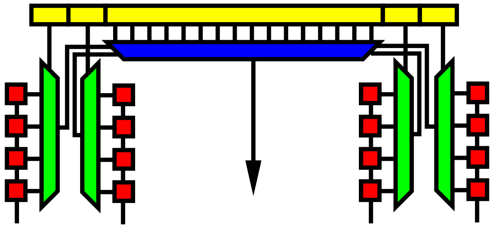
Retiming Structures

Some options for deeper retiming:



Retiming Structures

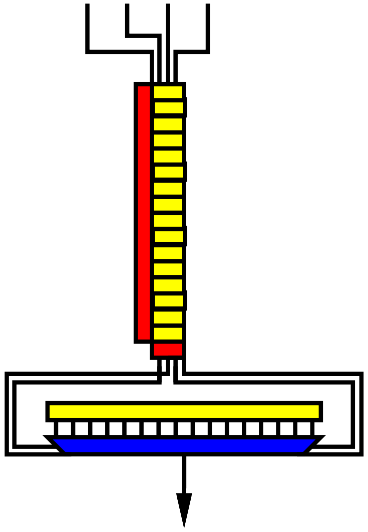
Flip-flop chain on inputs:



6-7Kλ^2 / flip-flop + select

Area	800Kλ^2 + 25-30Kλ^2 / stage
	900Kλ^2 ($d = 4$)
Bandwidth	1b/cycle
	1/d capacity

Retiming Structures



Register File:

From MIPS-X: 1Kλ^2/bit + 500λ^2/port

$$A_{RF} = (d+6)\cdot(W+6)\cdot\left(1\mathrm{K}\lambda^2 + p\cdot 500\lambda^2\right)$$

$w >> 6$, $d >> 6$ $\quad i+o=2$ **2K**λ^2/bit **(asymptotic)**

$w >> 6$, $d >> 6$ $\quad i=o=4$ **5K**λ^2/bit **(asymptotic)**

$w = 1$, $d >> 6$ $\quad i=o=4$ **35K**λ^2/bit **(asymptotic)**

$w = 1$, $d = 16$ $\quad i=o=4$ **770K**λ^2

Retiming Structures

Memory Blocks:
SRAM bit $\approx 1200\lambda^2$ (asymptotic)
DRAM bit $\approx 100\lambda^2$ (asymptotic)

Altera 10K (-3): 2048 bit block
Bandwidth 8b/cycle (8 ns cycle)
1/512 capacity

Retiming Structures

Disk drive:

- cheaper per bit than DRAM/flash
- Bandwidth: 10-20Mb/s
  For 10 ns array cycle $\rightarrow$ 1b/100 cycles

Memory Hierarchy

"Memory Hierarchy" arises from these area/bandwidth tradeoffs...

- Smaller/cheaper to store words/blocks (save routing, control)
- Smaller/cheaper to handle long-retiming in larger arrays (reduce routing requirements)
- High bandwidth out of registers / shallow memories

	DRAM	SRAM	RF bit	FF/RF	RF×1	XC	net FF	FF/LUT
λ^2	100	1200	2K	5K	40K	40K	200K	800K
bw/cap.	$1/10^7$	$1/10^5$–10^3		1/100	1/100	1/16	1/1	1/1

Retiming Requirements

- not unique to a problem
- depend on algorithm/implementation
- some behavioral transformation can alter significantly

Retiming Requirements Example

$$Q = A \cdot B + C \cdot D + E \cdot F$$

```
for i 1 to N
    t1[i] ← A[i]*B[i]
for i 1 to N
    t2[i] ← C[i]*D[i]
for i 1 to N
    t3[i] ← E[i]*F[i]
for i 1 to N
    t2[i] ← t1[i]+t2[i]
for i 1 to N
    Q[i] ← t2[i]+t3[i]
```

;; $3N$ registers

```
for i 1 to N
    t1←A[i]*B[i]
    t2←C[i]*D[i]
    t1←t1+t2
    t2←E[i]*F[i]
    Q[i]←t1+t2
```

;; 2 registers

Highlights

- Wide variety of retiming distances
- Wide variety of costs ($100\lambda^2 \rightarrow 10^6\lambda^2$)
- Routing and i/o bandwidth playing big factor in cost
- Give rise to memory hierarchy
- Retiming costs affected by high-level decisions/strategy in solving a task