ESE534: Computer Organization

Day 20: April 2, 2012
Retiming

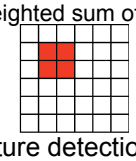
Today

- Retiming Demand
  - Folded Computation
  - Logical Pipelining
  - Physical Pipelining
- Retiming Supply
  - Technology
  - Structures
  - Hierarchy

Retiming Demand

Image Processing

- Many operations can be described in terms of 2D window filters
  - Compute value as a weighted sum of the neighborhood



- Blurring, edge and feature detection, object recognition and tracking, motion estimation, VLSI design rule checking

Preclass 2

- Describes window computation:

```
1: for x=0 to N-1
2:    for y=0 to N-1
3:       out[x][y]=0;
4:       for wx=0 to W-1
5:          for wy=0 to W-1
6:             out[x][y]+=in[x+wx][y+wy]*c[wx][wy]
```


Preclass 2

- How many times is each in[x][y] used?

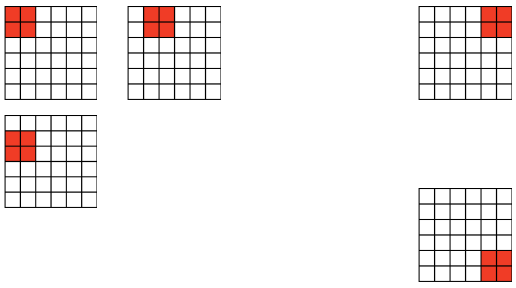
```
1: for x=0 to N-1
2:    for y=0 to N-1
3:       out[x][y]=0;
4:       for wx=0 to W-1
5:          for wy=0 to W-1
6:             out[x][y]+=in[x+wx][y+wy]*c[wx][wy]
```

Preclass 2

- Sequentialized on one multiplier,
  - Distance between in[x][y] uses?

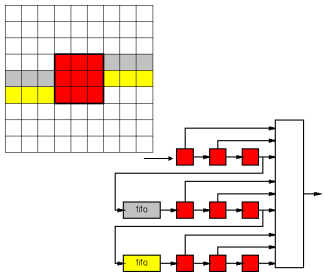
```
1: for x=0 to N-1
2:    for y=0 to N-1
3:       out[x][y]=0;
4:       for wx=0 to W-1
5:          for wy=0 to W-1
6:             out[x][y]+=in[x+wx][y+wy]*c[wx][wy]
```


Window



Parallel Scaling/Spatial Sum

- Compute entire window at once
  - More hardware, fewer cycles



Preclass 2

- Unroll inner loop,
  - Distance between in[x][y] uses?

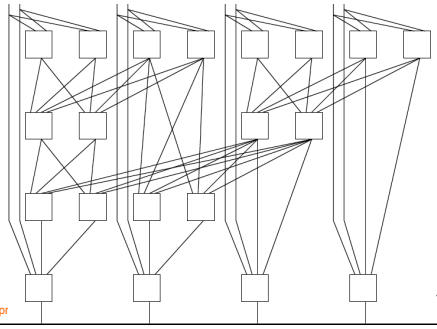
```
1: for x=0 to N-1
2:    for y=0 to N-1
3:       out[x][y]=0;
4:       for wx=0 to W-1
5:          for wy=0 to W-1
6:             out[x][y]+=in[x+wx][y+wy]*c[wx][wy]
```


Fully Spatial

- What if we gave each pixel its own processor?

Preclass 1

- How many registers on each link?



Flop Experiment #1

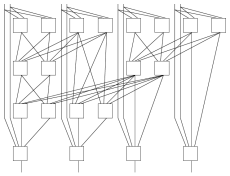
- Pipeline/C-slow/retime to single LUT delay per cycle
  - MCNC benchmarks to 256 4-LUTs
  - no interconnect accounting

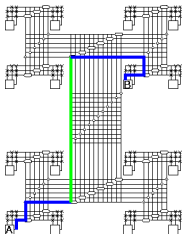
Number of Registers	1	2	3	4	5	6	7	8	9	10
Percentage	72	16	4.5	2.2	1.3	0.96	1.2	0.46	0.12	0.11

  - average 1.7 registers/LUT (some circuits 2--7)

Long Interconnect Path

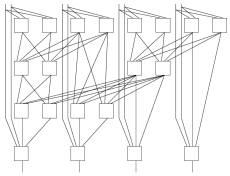
- What happens if one of these links ends up on a long interconnect path?

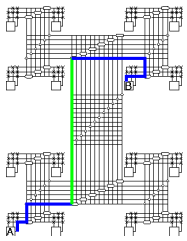




Pipeline Interconnect Path

- To avoid cycle being limited by longest interconnect
  - Pipeline network



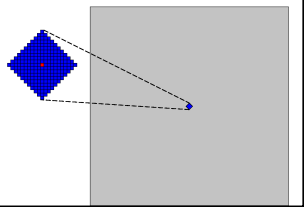


Chips >> Cycles

- Chips growing
- Gate delays shrinking
- Wire delays aren't scaling down
- ➔Will take many cycles to cross chip

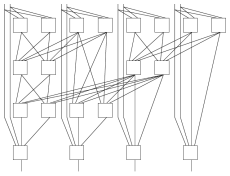
Clock Cycle Radius

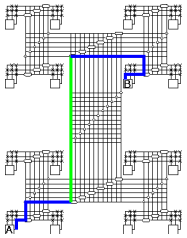
- Radius of logic can reach in one cycle (45 nm)
  - Radius 10 (preclass 19: L_{seg}=5 → 50ps)
    - Few hundred PEs
  - Chip side 600-700 PE
    - 400-500 thousand PEs
  - 100s of cycles to cross



Pipelined Interconnect

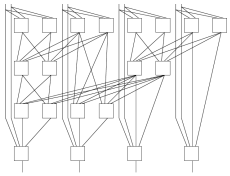
- In what cases is this convenient?

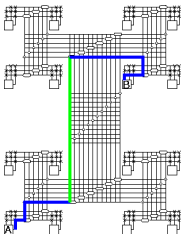




Pipelined Interconnect

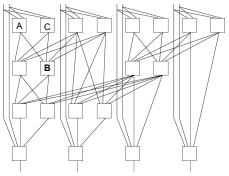
- When might it pose a challenge?

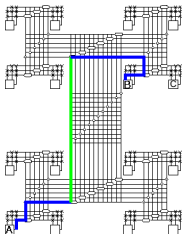




Long Interconnect Path

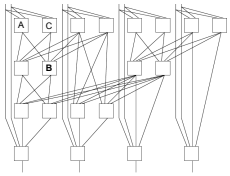
- What happens here?

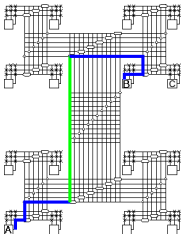




Long Interconnect Path

- Adds pipeline delays
- May force further pipelining of logic to balance out paths
- More registers





Reminder

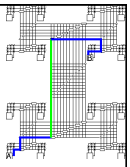
Flop Experiment #1

- Pipeline/C-slow/retime to single LUT delay per cycle
  - MCNC benchmarks to 256 4-LUTs
  - no interconnect accounting

Number of Registers	1	2	3	4	5	6	7	8	9	10
Percentage	72	16	4.5	2.2	1.3	0.96	1.2	0.46	0.12	0.11

  - average 1.7 registers/LUT (some circuits 2--7)

Flop Experiment #2



- Pipeline and retime to HSRA cycle
  - place on HSRA
  - single LUT or interconnect timing domain
  - same MCNC benchmarks

Number of Registers	1	2	3	4	5	6	7	8	9	10	>10
Percentage	60	6.9	5.9	3.8	4.3	2.7	2.6	1.9	1.5	1.2	9.2

  - average 4.7 registers/LUT

[Tsu et al., FPGA 1999]

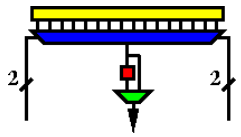
Retiming Requirements

- Retiming requirement depends on parallelism and performance
- Even with a given amount of parallelism
  - Will have a distribution of retiming requirements
  - May differ from task to task
  - May vary **independently** from compute/ interconnect requirements
    - Another balance issue to watch
      - Balance with compute, interconnect
    - Need a canonical way to measure
      - Like Rent?

Retiming Supply

Optional Output

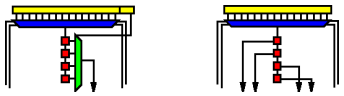
- Flip-flop (optionally) on output



  - flip-flop: 1K F^2
  - Switch to select: ~ 1.25K F^2
  - Area: 1 LUT (200K→250K F^2/LUT)
  - Bandwidth: 1b/cycle

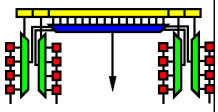
Output

- Single Output
  - Ok, if don't need other timings of signal
- Multiple Output
  - more routing



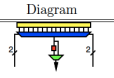
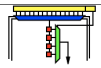
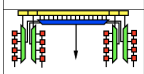
Input

- More registers (K×)
  - 2.5K F^2/register+mux
  - 4-LUT => 10K F^2/depth



- No more interconnect than unretimed
  - ***open***: compare savings to additional reg. cost
  - Area: 1 LUT (250K+d*10K F^2) get Kd regs
    - d=4, 290K F^2
  - Bandwidth: K/cycle
    - 1/d th capacity

Preclass 3

Diagram	Description	Area per Block	Blocks Needed	Total Area	Best?
	4-LUT with single optional flip-flop on output	252.5K F^2			
	4-LUT with configurable depth output register with maximum depth 4	260K F^2			
	4-LUT with configurable depth input registers with maximum depth 4	290K F^2			

Day 7

Some Numbers (memory)

- Unit of area = F^2 ($F=2\lambda$)
- Register as stand-alone element ≈ 1000 F^2
  - *e.g.* as needed/used Day 4
- Static RAM cell ≈ 250 F^2
  - SRAM Memory (single ported)
- Dynamic RAM cell (DRAM process) ≈ 25 F^2
- Dynamic RAM cell (SRAM process) ≈ 75 F^2

Retiming Density

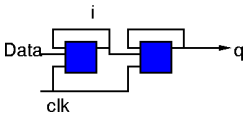
- LUT+interconnect ≈ 250K F^2
- Register as stand-alone element ≈ 1K F^2
- Static RAM cell ≈ 250 F^2
  - – SRAM Memory (single ported)
- Dynamic RAM cell (DRAM process) ≈ 25 F^2
- Dynamic RAM cell (SRAM process) ≈ 75 F^2
- Can have much more retiming memory per chip if put it in large arrays
  - – …but then cannot get to it as frequently

Retiming Structure Concerns

- Area: F^2/bit
- Throughput: bandwidth (bits/time)
- Energy

Just Logic Blocks

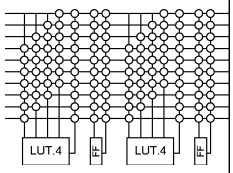
- Most primitive
  - – build flip-flop out of logic blocks
    - I ←D*/Clk + I*Clk
    - Q ←Q*/Clk + I*Clk



  - – Area: 2 LUTs (200K→250K F^2/LUT each)
  - – Bandwidth: 1b/cycle

Separate Flip-Flops

- Network flip flop w/ own interconnect
  - + can deploy where needed
  - − requires more interconnect
  - + Vary LUT/FF ratio
    - Arch. Parameter
  - Assume routing ∝ inputs
    - 1/4 size of LUT
  - Area: 50K F^2 each
  - Bandwidth: 1b/cycle



Virtex SRL16

- Xilinx Virtex 4-LUT
  - – Use as 16b shiftreg

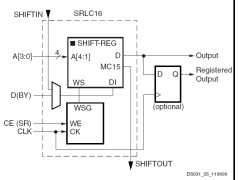
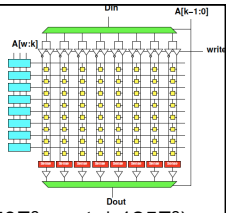


Figure 21: **Shift Register Configurations**

- Area: ~250K F^2/16≈16K F^2/bit
  - – Does not need CLBs to control
- Bandwidth: 1b/2 cycle (1/2 CLB)
  - – 1/16 th capacity

Register File Memory Bank

- From MIPS-X
  - – 250F^2/bit + 125F^2/port
  - – Area(RF) = (d+6)(W+6)(250F^2+ports* 125F^2)

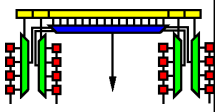


Preclass 4

- Complete Table
- How small can get?
- Compare w=1, p=8 case to input retiming

Input

- More registers (K×)
  - 2.5K F^2/register+mux
  - 4-LUT => 10K F^2/depth



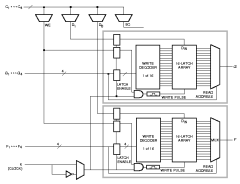
- No more interconnect than unretimed
  - ***open***: compare savings to additional reg. cost
  - Area: 1 LUT (1M+d*10K F^2) get Kd regs
    - d=4, 290K F^2
  - Bandwidth: K/cycle
    - 1/d th capacity

Preclass 4

- Note compactness from wide words (share decoder)

Xilinx CLB

- Xilinx 4K CLB
  - as memory
  - works like RF



- Area: 1/2 CLB (160K F^2)/16≈10K F^2/bit
  - but need 4 CLBs to control
- Bandwidth: 1b/2 cycle (1/2 CLB)
  - 1/16 th capacity

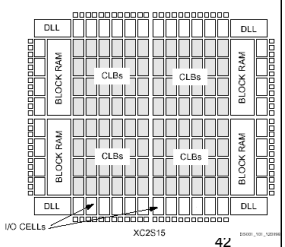
Memory Blocks

- SRAM bit ≈ 300 F^2 (large arrays)
- DRAM bit ≈ 25 F^2 (large arrays)

- Bandwidth: W bits / 2 cycles
  - usually single read/write
  - $1/2^A$ th capacity

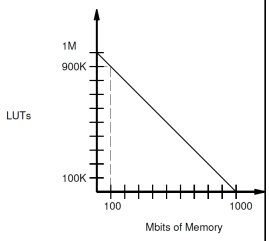
Dual-Ported Block RAMs

- Virtex-6 Series 36Kb memories
- Stratix-4
  - 640b, 9Kb, 144Kb
- Can put 250K/250≈1K bits in space of 4-LUT
  - Trade few 4-LUTs for considerable memory



Dual-Ported Block RAMs

- Virtex-6 Series 36Kb memories
- Stratix-4
  - 640b, 9Kb, 144Kb
- Can put 250K/250≈1K bits in space of 4-LUT
  - Trade few 4-LUTs for considerable memory



Hierarchy/Structure Summary

- Big Idea: "Memory Hierarchy" arises from area/bandwidth tradeoffs
  - Smaller/cheaper to store words/blocks
    - (saves routing and control)
  - Smaller/cheaper to handle long retiming in larger arrays (reduce interconnect)
  - High bandwidth out of shallow memories
  - Applications have mix of retiming needs

(Area, BW) → Hierarchy

	Area (F^2)	Bw/capacity
FF/LUT	250K	1/1
netFF	50K	1/1
XC	10K	1/16
RFx1	10K	1/100
FF/RF	1K	1/100
RF bit	2K	1/100
SRAM	300	$1/10^5$
DRAM	25	$1/10^7$

Modern FPGAs

- Output Flop (depth 1)
- Use LUT as Shift Register (16)
- Embedded RAMs (9Kb,36Kb)
- Interface off-chip DRAM (~0.1—1Gb)
- No retiming in interconnect
  - ….yet

Modern Processors

- DSPs have accumulator (depth 1)
- Inter-stage pipelines (depth 1)
  - Lots of pipelining in memory path…
- Reorder Buffer (4—32)
- Architected RF (16, 32, 128)
- Actual RF (256, 512…)
- L1 Cache (~64Kb)
- L2 Cache (~1Mb)
- L3 Cache (10-100Mb)
- Main Memory in DRAM (~10-100Gb)

Admin

- Final out now
  - 1 month exercise
  - Milestone deadlines next two Mondays
- Wednesday Lossless compression?
  - Final assignment + cited papers for Wed?
  - OR Reading for Wednesday on web?
  - Weigh in on feedback sheet

Big Ideas [MSB Ideas]

- Tasks have a wide variety of retiming distances (depths)
  - Within design, among tasks
- Retiming requirements vary independently of compute, interconnect requirements (balance)
- Wide variety of retiming costs
  - 25 F^2→250K F^2
- Routing and I/O bandwidth
  - big factors in costs
- Gives rise to memory (retiming) hierarchy