ESE535: Electronic Design Automation

Day 3: January 26, 2009
Scheduled Operator Sharing

Last Time

• How to construct a dataflow graph from a high-level language
  – From C

Today

• Sharing Resources
• Area-Time Tradeoffs
• Throughput vs. Latency
• VLIW Architectures

Compute Function

• Compute:
  \[ y = Ax^2 + Bx + C \]
• Assume
  – D(Mpy) > D(Add)
  – A(Mpy) > A(Add)

Spatial Quadratic

• \[ A(\text{Quad}) = 3 \times A(Mpy) + 2 \times A(Add) \]

Latency vs. Throughput

• Latency: Delay from inputs to output(s)
• Throughput: Rate at which can introduce new set of inputs
Washer/Dryer Example

- 1 Washer Takes 30 minutes
- 1 Dryer Takes 45 minutes
- How long to do one load of wash? → Wash latency
- How long to do 5 loads of wash?
- Wash Throughput?

Spatial Quadratic

- \( D(\text{Quad}) = 2D(\text{Mpy})+D(\text{Add}) \)
- Throughput \( 1/(2D(\text{Mpy})+D(\text{Add})) \)
- \( A(\text{Quad}) = 3A(\text{Mpy}) + 2A(\text{Add}) \)

Pipelined Spatial Quadratic

- \( D(\text{Quad}) = 3D(\text{Mpy}) \)
- Throughput \( 1/D(\text{Mpy}) \)
- \( A(\text{Quad}) = 3A(\text{Mpy}) + 2A(\text{Add})+6A(\text{Reg}) \)

Quadratic with Single Multiplier and Adder?

- We’ve seen reuse to perform the same operation
  → pipelining
- We can also reuse a resource in time to perform a different role.
  → Here: \( x\times x, A\times(x \times x), B \times x \)
  → also: \( (Bx)+c, (A \times x \times x)+(Bx+c) \)

Quadratic Datapath

- Start with one of each operation

Multiplier serves multiple roles
- \( x \times x \)
- \( A \times (x \times x) \)
- \( B \times x \)
- Will need to be able to steer data (switch interconnections)
Quadratic Datapath

- Multiplier serves multiple roles
  - \(x \times x\)
  - \(A \times (x \times x)\)
  - \(B \times x\)
- \(x, x \times x\)
- \(x, A, B\)

Quadratic Datapath

- Adder serves multiple roles
  - \((B \times x) + c\)
  - \((A \times x \times x) + (B \times x + c)\)
- one always mpy output
- \(C, B \times x + C\)

Quadratic Datapath

- Add input register for \(x\)

Cycle Impact?

- Add mux delay
- Register setup/hold time, clock skew
- Limited by slowest operation
Quadratic Control

• Now, we just need to control the datapath
• What control?
• Control:
  - LD x
  - LD x*x
  - MA Select
  - MB Select
  - AB Select
  - LD Bx+C
  - LD Y

Quadratic Memory Control

1. LD_X
2. MA_SEL=x, MB_SEL[1:0]=x, LD_x*x
3. MA_SEL=x, MB_SEL[1:0]=B
4. AB_SEL=C, MA_SEL=x*x, MB_SEL=A, LD_Bx+C
5. AB_SEL=Bx+C, LD_Y

Quadratic Datapath

• Latency/Throughput/Area?
• Latency: 5*(D(MPY)+D(mux3))
• Throughput: 1/Latency
• Area: A(Mpy)+A(Add)+5*A(Reg) +2*A(Mux2)+A(Mux3)+A(Imem)

Registers → Memory

• Generally can see many registers
• If # registers >> physical operators
  – Only need to access a few at a time
• Group registers into memory banks

Memory Bank Quadratic

• Store x
• x*x
• B*x
• A*x^2; B*x+c
• (A*x^2)+(B*x+c)
Memory Bank Quadratic

- Store \( x \)
- \( x^2 \)
- \( B^x \)
- \( A^x^2; B^x+c \)
- \( (A^x^2)+(B^x+c) \)

Cycle Impact?

- Add mux delay
- Register setup/hold time, clock skew
- Memory read/write
  - Could pipeline
  - Impact?
    - Latency
    - Throughput?
- Limited by slowest operation

VLIW

- Very Long Instruction Word
- Set of operators
  - Parameterize number, distribution
  - Gives rise to Area-Time tradeoff
  - Fewer operators \( \rightarrow \) less time, more area
  - More operators \( \rightarrow \) more time, less area
- Memories for intermediate state
- Memory for "long" instructions
- Schedule compute task
- General, potentially more expensive than customized
  - Wiring, memories get expensive
  - Opportunity for further optimizations

Summary

- Reuse physical operators in time
- Share operators in different roles
- Allows us to reduce area at expense of increasing time
- Area-Time tradeoff
- Pay some sharing overhead
  - Muxes, memory
- VLIW – general formulation for shared datapaths

Admin

- Reading for Wednesday
Big Ideas:

• Scheduled Operator Sharing