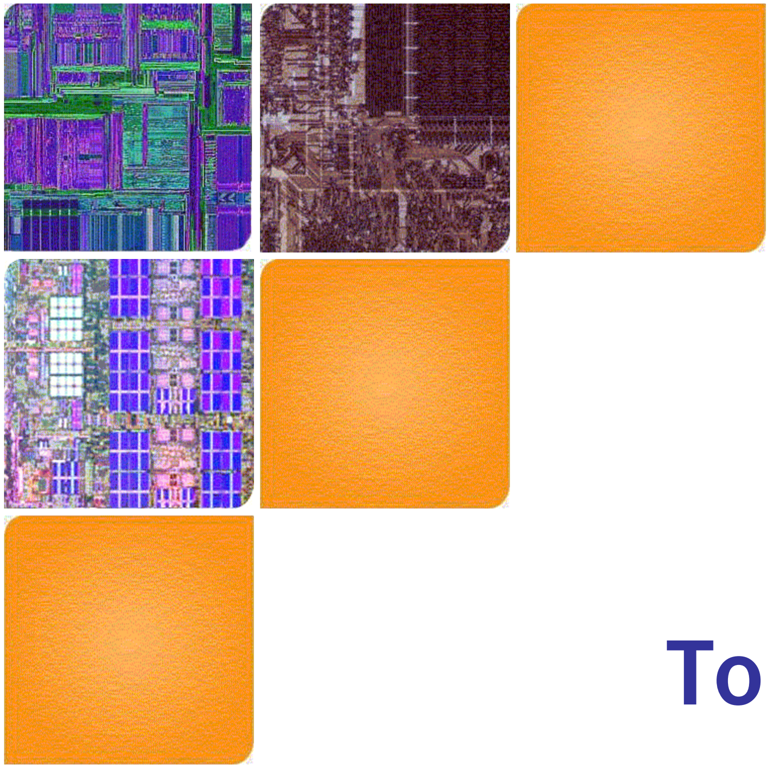


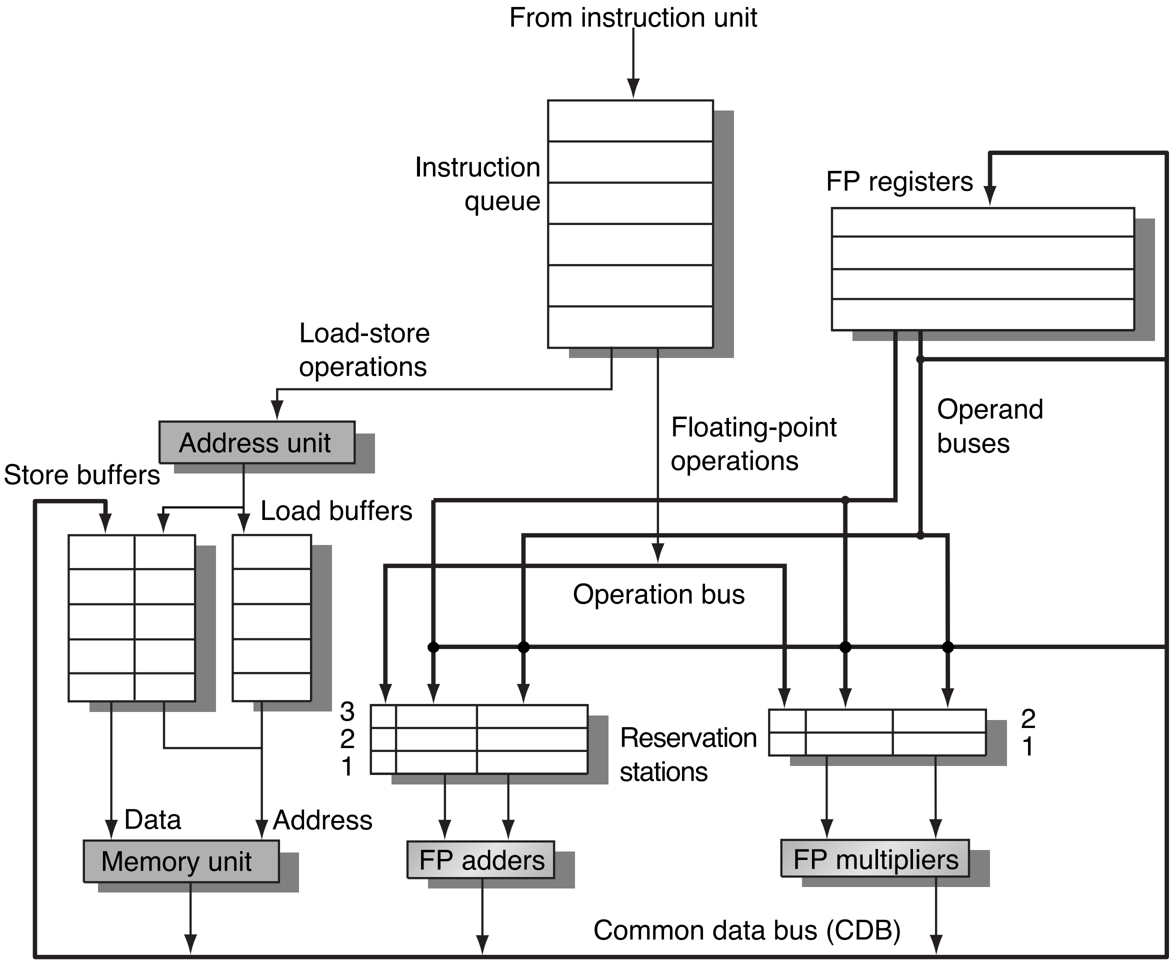
# CS6290
# Tomasulo's Algorithm

Georgia Tech College of Computing

# Implementing Dynamic Scheduling

- Tomasulo’s Algorithm
  - Used in IBM 360/91 (in the 60s)
  - Tracks when operands are available to satisfy data dependences
  - Removes name dependences through register renaming
  - Very similar to what is used today
    - Almost all modern high-performance processors use a derivative of Tomasulo’s… much of the terminology survives to today.

# Tomasulo's Algorithm: The Picture



# Issue (1)

- Get next instruction from instruction queue.
- Find a free *reservation station* for it
  (if none are free, stall until one is)
- Read operands that are in the registers
- If the operand is not in the register,
  find which reservation station will produce it
- In effect, this step renames registers
  (reservation station IDs are “temporary” names)

# Issue (2)

Instruction Buffers

|  |  |
|---|---|
| 3. | F1 = F2 + F3 |
| 2. | F4 = F1 – F2 |
| 1. | F1 = F2 / F3 |

<u>To-Do list (from last slide)</u>:
Get next inst from IB's
Find free reservation station
Read operands from RF
Record source of other operands
Update source mapping (RAT)

Reg File

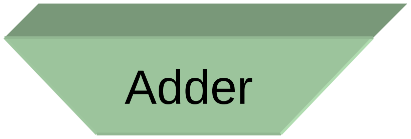
|  |  |
|---|---|
| F1 | 3.141593 |
| F2 | -1.00000 |
| F3 | 2.718282 |
| F4 | 0.707107 |

RAT

|  |  |
|---|---|
| F1 | 0 |
| F2 | 1 |
| F3 | 0 |
| F4 | 0 |

|  |  |  |  |
|---|---|---|---|
| A1 (1) | F2=F4+F1 | 0.7071 | π |
| A2 (2) |  |  |  |
| A3 (3) |  |  |  |

|  |  |  |  |
|---|---|---|---|
| C1 (4) | F1=F2/F3 | 1(A1) | 2.718 |
| C2 (5) |  |  |  |



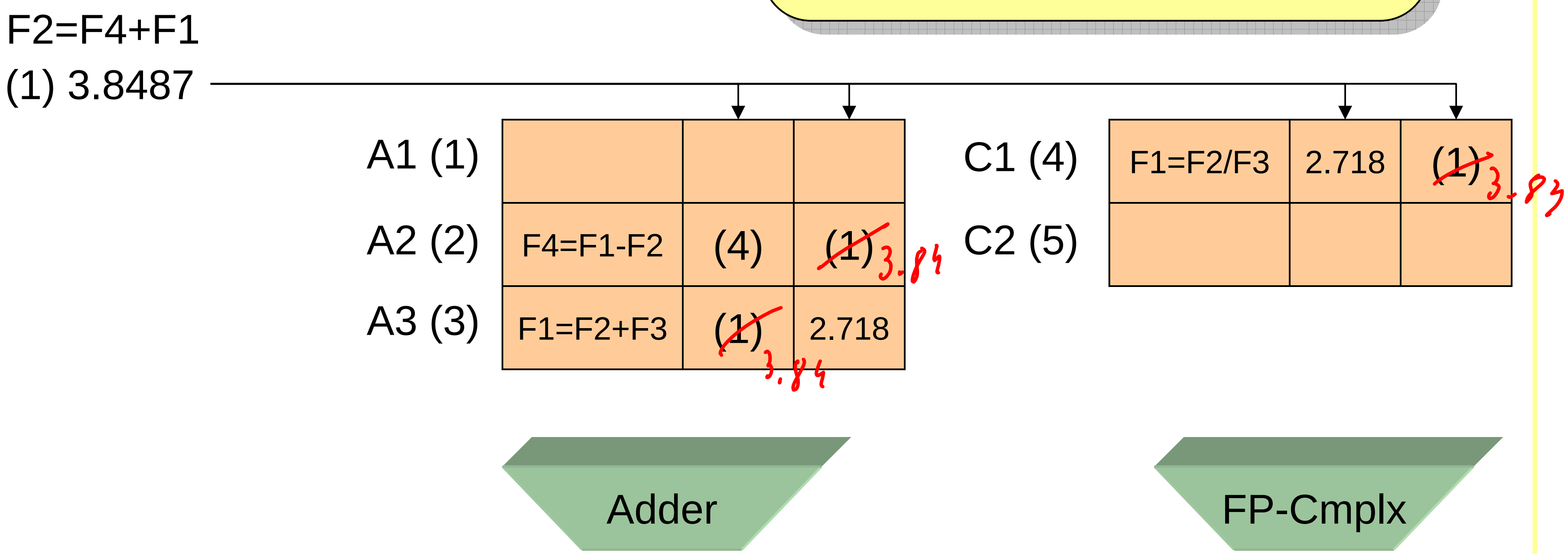
Georgia Tech College of Computing

# Execute (1)

- Monitor results as they are produced
- Put a result into all reservation stations waiting for it (missing source operand)
- When all operands available for an instruction, it is ready (we can actually execute it)
- Several ready instrs for one functional unit?
  - Pick one.
  - Except for load/store
    Load/Store must be done in
    the proper order to avoid hazards through memory
    (more loads/stores this in a later lecture)

# Execute (2)

To-Do list (from last slide):
Monitor results from ALUs
Capture matching operands
Compete for ALUs



F2=F4+F1
(1) 3.8487

|  |  |  |  |
|---|---|---|---|
| A1 (1) |  |  |  |
| A2 (2) | F4=F1-F2 | (4) | ~~(1)~~ 3.84 |
| A3 (3) | F1=F2+F3 | ~~(1)~~ 3.84 | 2.718 |

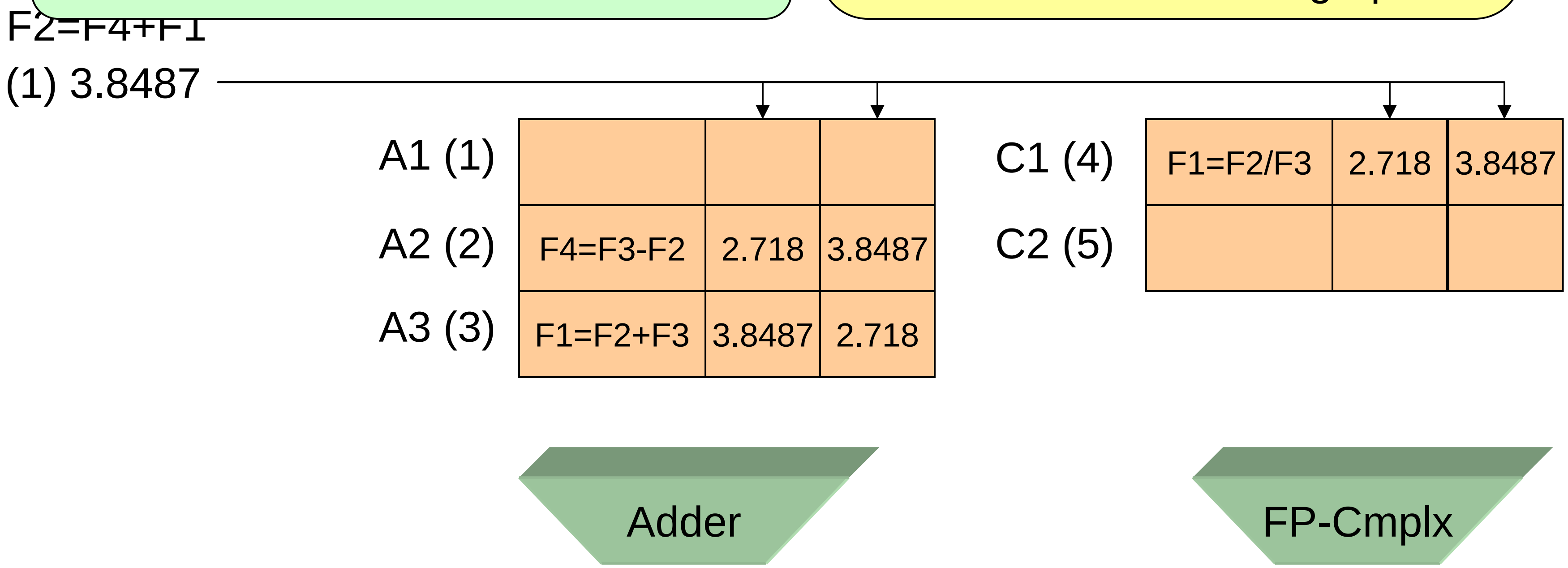
|  |  |  |  |
|---|---|---|---|
| C1 (4) | F1=F2/F3 | 2.718 | ~~(1)~~ 3.83 |
| C2 (5) |  |  |  |

# Execute (3)
# More than one ready inst for the same unit

Common heuristic: oldest first

You can do whatever: it only affects performance, not correctness

Optimal is impossible:
Precedence constrained scheduling problem is NP-complete [GJ,p239]
… and that assumes you have access to the entire graph


F2=F4+F1
(1) 3.8487

| | | | |
|---|---|---|---|
| A1 (1) | | | |
| A2 (2) | F4=F3-F2 | 2.718 | 3.8487 |
| A3 (3) | F1=F2+F3 | 3.8487 | 2.718 |

| | | | |
|---|---|---|---|
| C1 (4) | F1=F2/F3 | 2.718 | 3.8487 |
| C2 (5) | | | |

# Write Result (1)

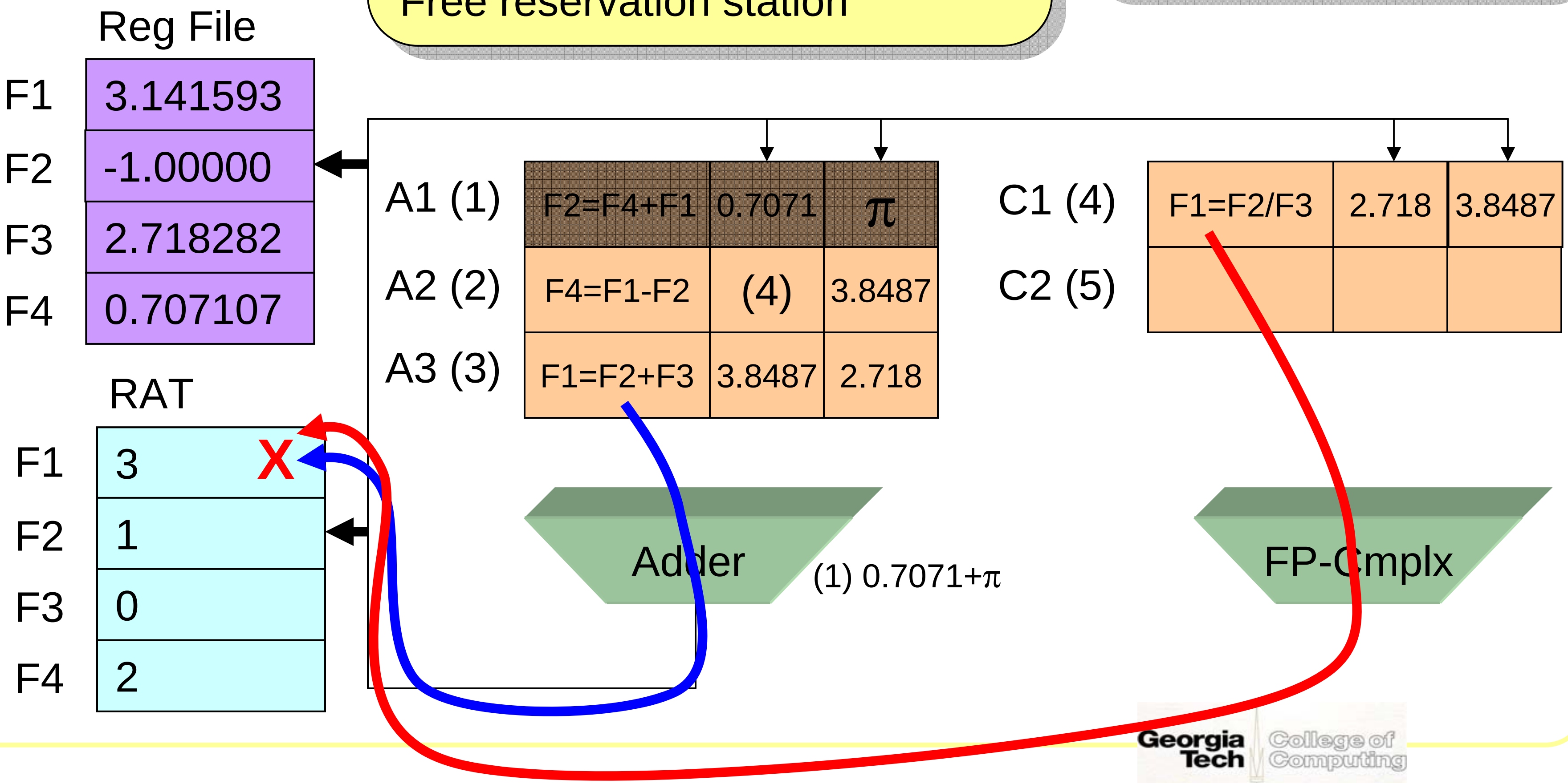
- When result is computed, make it available on the “common data bus” (CDB), where waiting reservation stations can pick it up
- Stores write to memory
- Result stored in the register file
- This step frees the reservation station
- For our register renaming, this recycles the temporary name (future instructions can again find the value in the actual register, until it is renamed again)

# Write Result (2)

0. F2 = F4 + F1
1. F1 = F2 / F3
2. F4 = F1 – F2
3. F1 = F2 + F3

<u>To-Do list (from last slide)</u>:
Broadcast on CDB
Writeback to RF
Update Mapping
Free reservation station

Only update RAT (and RF) if RAT still contains your mapping!



Reg File

| | |
|---|---|
| F1 | 3.141593 |
| F2 | -1.00000 |
| F3 | 2.718282 |
| F4 | 0.707107 |

RAT

| | |
|---|---|
| F1 | 3 X |
| F2 | 1 |
| F3 | 0 |
| F4 | 2 |

| | | | |
|---|---|---|---|
| A1 (1) | F2=F4+F1 | 0.7071 | π |
| A2 (2) | F4=F1-F2 | (4) | 3.8487 |
| A3 (3) | F1=F2+F3 | 3.8487 | 2.718 |

| | | | |
|---|---|---|---|
| C1 (4) | F1=F2/F3 | 2.718 | 3.8487 |
| C2 (5) | | | |

# Tomasulo’s Algorithm: Load/Store

- The reservation stations take care of dependences through registers.
- Dependences also possible through memory
  - Loads and stores not reordered in original IBM 360
  - We’ll talk about how to do load-store reordering later

# Detailed Example

Load: 2 cycles
Add: 2 cycles
Mult: 10 cycles
Divide: 40 cycles

Assume

R2 is 100

R3 is 200

F4 is 2.5

| | | Is | Ex | W |
|---|---|---|---|---|
| 1. L.D | F6, 34(R2) | 1 | | |
| 2. L.D | F2, 45(R3) | | | |
| 3. MUL.D | F0, F2, F4 | | | |
| 4. SUB.D | F8, F2, F6 | | | |
| 5. DIV.D | F10,F0,F6 | | | |
| 6. ADD.D | F6, F8, F2 | | | |

**Reservation Stations**

| | Busy | Op | Vj | Vk | Qj | Qk | A |
|---|---|---|---|---|---|---|---|
| LD1 | 1 | L.D | | | | | 134 |
| LD2 | | | | | | | |
| AD1 | | | | | | | |
| AD2 | | | | | | | |
| AD3 | | | | | | | |
| ML1 | | | | | | | |
| ML2 | | | | | | | |

Cycle: 1

Register Status:

| F0 | F2 | F4 | F6 | F8 | F10 | F12 | |
|---|---|---|---|---|---|---|---|
| | | | LD1 | | | | … |

Georgia Tech College of Computing

# Detailed Example

Load: 2 cycles
Add: 2 cycles
Mult: 10 cycles
Divide: 40 cycles

Assume

R2 is 100

R3 is 200

F4 is 2.5

| | | Is | Ex | W |
|---|---|---|---|---|
| 1. L.D | F6, 34(R2) | 1 | 2 | |
| 2. L.D | F2, 45(R3) | 2 | | |
| 3. MUL.D | F0, F2, F4 | | | |
| 4. SUB.D | F8, F2, F6 | | | |
| 5. DIV.D | F10,F0,F6 | | | |
| 6. ADD.D | F6, F8, F2 | | | |

**Reservation Stations**

| | Busy | Op | Vj | Vk | Qj | Qk | A |
|---|---|---|---|---|---|---|---|
| LD1 | 1 | L.D | | | | | 134 |
| LD2 | 1 | L.D | | | | | 245 |
| AD1 | | | | | | | |
| AD2 | | | | | | | |
| AD3 | | | | | | | |
| ML1 | | | | | | | |
| ML2 | | | | | | | |

Register Status:

| F0 | F2 | F4 | F6 | F8 | F10 | F12 | |
|---|---|---|---|---|---|---|---|
| | LD2 | | LD1 | | | | … |

Cycle: 2

# Detailed Example

Load: 2 cycles
Add: 2 cycles
Mult: 10 cycles
Divide: 40 cycles

Assume

R2 is 100

R3 is 200

F4 is 2.5

| | | Is | Ex | W |
|---|---|---|---|---|
| 1. L.D | F6, 34(R2) | 1 | 4 | |
| 2. L.D | F2, 45(R3) | 2 | 3 | |
| 3. MUL.D | F0, F2, F4 | 3 | | |
| 4. SUB.D | F8, F2, F6 | | | |
| 5. DIV.D | F10,F0,F6 | | | |
| 6. ADD.D | F6, F8, F2 | | | |

**Reservation Stations**

| | Busy | Op | Vj | Vk | Qj | Qk | A |
|---|---|---|---|---|---|---|---|
| LD1 | 1 | L.D | | | | | 134 |
| LD2 | 1 | L.D | | | | | 245 |
| AD1 | 1 | MUL.D | | 2.4 | LD2 | | |
| AD2 | | | | | | | |
| AD3 | | | | | | | |
| ML1 | | | | | | | |
| ML2 | | | | | | | |

Cycle: 3

Register Status:

| F0 | F2 | F4 | F6 | F8 | F10 | F12 | |
|---|---|---|---|---|---|---|---|
| AD1 | LD2 | | LD1 | | | | … |

# Detailed Example

Load: 2 cycles
Add: 2 cycles
Mult: 10 cycles
Divide: 40 cycles

Assume

R2 is 100

R3 is 200

F4 is 2.5

| | | Is | Ex | W |
|---|---|---|---|---|
| 1. L.D | F6, 34(R2) | 1 | 2 | 4 |
| 2. L.D | F2, 45(R3) | 2 | 3 | |
| 3. MUL.D | F0, F2, F4 | 3 | | |
| 4. SUB.D | F8, F2, F6 | 4 | | |
| 5. DIV.D | F10,F0,F6 | | | |
| 6. ADD.D | F6, F8, F2 | | | |

**Reservation Stations**

| | Busy | Op | Vj | Vk | Qj | Qk | A |
|---|---|---|---|---|---|---|---|
| LD1 | ~~1~~ 0 | L.D | | | | | 134 |
| LD2 | 1 | L.D | | | | | 245 |
| AD1 | 1 | SUB.D | | Val | LD2 | ~~LD1~~ | |
| AD2 | | | | | | | |
| AD3 | | | | | | | |
| ML1 | 1 | MUL.D | | 2.5 | LD2 | | |
| ML2 | | | | | | | |

Cycle: 4

Register Status:

| F0 | F2 | F4 | F6 | F8 | F10 | F12 | |
|---|---|---|---|---|---|---|---|
| ML1 | LD2 | | ~~LD1~~ | AD1 | | | … |

# Detailed Example

Load: 2 cycles
Add: 2 cycles
Mult: 10 cycles
Divide: 40 cycles

Assume

R2 is 100

R3 is 200

F4 is 2.5

| | | Is | Ex | W |
|---|---|---|---|---|
| 1. L.D | F6, 34(R2) | 1 | 2 | 4 |
| 2. L.D | F2, 45(R3) | 2 | 3 | 6 |
| 3. MUL.D | F0, F2, F4 | 3 | | |
| 4. SUB.D | F8, F2, F6 | 4 | | |
| 5. DIV.D | F10,F0,F6 | 5 | | |
| 6. ADD.D | F6, F8, F2 | | | |

**Reservation Stations**

| | Busy | Op | Vj | Vk | Qj | Qk | A |
|---|---|---|---|---|---|---|---|
| LD1 | 0 | | | | | | |
| LD2 | ~~1~~ 0 | L.D | | | | | 245 |
| AD1 | 1 | SUB.D | val | 0.5 | ~~LD2~~ | | |
| AD2 | | | | | | | |
| AD3 | | | | | | | |
| ML1 | 1 | MUL.D | val | 2.5 | ~~LD2~~ | | |
| ML2 | 1 | DIV.D | | | ML1 | ~~LD2~~ ML2 | |

Cycle: 5

Register Status:

| F0 | F2 | F4 | F6 | F8 | F10 | F12 | |
|---|---|---|---|---|---|---|---|
| ML1 | ~~LD2~~ | | ML2 | AD1 | | | … |

# Detailed Example

Load: 2 cycles
Add: 2 cycles
Mult: 10 cycles
Divide: 40 cycles

Assume

R2 is 100

R3 is 200

F4 is 2.5

| Instruction | | Is | Ex | W |
|---|---|---|---|---|
| 1. L.D | F6, 34(R2) | 1 | 2 | 4 |
| 2. L.D | F2, 45(R3) | 2 | 3 | 5 |
| 3. MUL.D | F0, F2, F4 | 3 | 6 | |
| 4. SUB.D | F8, F2, F6 | 4 | 6 | |
| 5. DIV.D | F10,F0,F6 | 5 | | |
| 6. ADD.D | F6, F8, F2 | 6 | | |

**Reservation Stations**

| | Busy | Op | Vj | Vk | Qj | Qk | A |
|---|---|---|---|---|---|---|---|
| LD1 | 0 | | | | | | |
| LD2 | 0 | | | | | | |
| AD1 | 1 | SUB.D | 1.5 | 0.5 | | | |
| AD2 | 1 | ADD.D | | val | AD1 | | |
| AD3 | | | | | | | |
| ML1 | 1 | MUL.D | 1.5 | 2.5 | | | |
| ML2 | 1 | DIV.D | | 0.5 | ML1 | | |

Cycle: 6

Register Status:

| F0 | F2 | F4 | F6 | F8 | F10 | F12 | |
|---|---|---|---|---|---|---|---|
| ML1 | | | AD2 | AD1 | ML2 | | … |

# Detailed Example

Load: 2 cycles
Add: 2 cycles
Mult: 10 cycles
Divide: 40 cycles

Assume

R2 is 100

R3 is 200

F4 is 2.5

| | | Is | Ex | W |
|---|---|---|---|---|
| 1. L.D | F6, 34(R2) | 1 | 2 | 4 |
| 2. L.D | F2, 45(R3) | 2 | 3 | 5 |
| 3. MUL.D | F0, F2, F4 | 3 | 6 | |
| 4. SUB.D | F8, F2, F6 | 4 | 6 | 8 |
| 5. DIV.D | F10,F0,F6 | 5 | | |
| 6. ADD.D | F6, F8, F2 | 6 | | |

**Reservation Stations**

| | Busy | Op | Vj | Vk | Qj | Qk | A |
|---|---|---|---|---|---|---|---|
| LD1 | 0 | | | | | | |
| LD2 | 0 | | | | | | |
| AD1 | ~~1~~ 0 | SUB.D | 1.5 | 0.5 | | | |
| AD2 | 1 | ADD.D | 1.0 | 2.5 | ~~AD1~~ | | |
| AD3 | | | | | | | |
| ML1 | 1 | MUL.D | 1.5 | 2.5 | | | |
| ML2 | 1 | DIV.D | | 0.5 | ML1 | | |

Cycle: 8

Register Status:

| F0 | F2 | F4 | F6 | F8 | F10 | F12 | |
|---|---|---|---|---|---|---|---|
| ML1 | | | AD2 | ~~AD1~~ | ML2 | | … |

# Detailed Example

Load: 2 cycles
Add: 2 cycles
Mult: 10 cycles
Divide: 40 cycles

Assume

R2 is 100

R3 is 200

F4 is 2.5

| | | Is | Ex | W |
|---|---|---|---|---|
| 1. L.D | F6, 34(R2) | 1 | 2 | 4 |
| 2. L.D | F2, 45(R3) | 2 | 3 | 5 |
| 3. MUL.D | F0, F2, F4 | 3 | 6 | |
| 4. SUB.D | F8, F2, F6 | 4 | 6 | 8 |
| 5. DIV.D | F10,F0,F6 | 5 | | |
| 6. ADD.D | F6, F8, F2 | 6 | 9 | |

**Reservation Stations**

| | Busy | Op | Vj | Vk | Qj | Qk | A |
|---|---|---|---|---|---|---|---|
| LD1 | 0 | | | | | | |
| LD2 | 0 | | | | | | |
| AD1 | 0 | | | | | | |
| AD2 | 1 | ADD.D | 1.0 | 2.5 | | | |
| AD3 | | | | | | | |
| ML1 | 1 | MUL.D | 1.5 | 2.5 | | | |
| ML2 | 1 | DIV.D | | 0.5 | ML1 | | |

Cycle: 9

Register Status:

| F0 | F2 | F4 | F6 | F8 | F10 | F12 | |
|---|---|---|---|---|---|---|---|
| ML1 | | | AD2 | | ML2 | | … |

# Detailed Example

Load: 2 cycles
Add: 2 cycles
Mult: 10 cycles
Divide: 40 cycles

Assume

R2 is 100

R3 is 200

F4 is 2.5

|  |  |  | Is | Ex | W |
|---|---|---|---|---|---|
| 1. | L.D | F6, 34(R2) | 1 | 2 | 4 |
| 2. | L.D | F2, 45(R3) | 2 | 3 | 5 |
| 3. | MUL.D | F0, F2, F4 | 3 | 6 |  |
| 4. | SUB.D | F8, F2, F6 | 4 | 6 | 8 |
| 5. | DIV.D | F10,F0,F6 | 5 |  |  |
| 6. | ADD.D | F6, F8, F2 | 6 | 9 | 11 |

**Reservation Stations**

|  | Busy | Op | Vj | Vk | Qj | Qk | A |
|---|---|---|---|---|---|---|---|
| LD1 | 0 |  |  |  |  |  |  |
| LD2 | 0 |  |  |  |  |  |  |
| AD1 | 0 |  |  |  |  |  |  |
| AD2 | ~~1~~ 0 | ADD.D | 1.0 | 2.5 |  |  |  |
| AD3 |  |  |  |  |  |  |  |
| ML1 | 1 | MUL.D | 1.5 | 2.5 |  |  |  |
| ML2 | 1 | DIV.D |  | 0.5 | ML1 |  |  |

Cycle: 11

| Register Status: | F0 | F2 | F4 | F6 | F8 | F10 | F12 |  |
|---|---|---|---|---|---|---|---|---|
|  | ML1 |  |  | ~~AD2~~ |  | ML2 |  | … |

# Detailed Example

Load: 2 cycles
Add: 2 cycles
Mult: 10 cycles
Divide: 40 cycles

Assume

R2 is 100

R3 is 200

F4 is 2.5

| | Is | Ex | W |
|---|---|---|---|
| 1. L.D F6, 34(R2) | 1 | 2 | 4 |
| 2. L.D F2, 45(R3) | 2 | 3 | 5 |
| 3. MUL.D F0, F2, F4 | 3 | 6 | 16 |
| 4. SUB.D F8, F2, F6 | 4 | 6 | 8 |
| 5. DIV.D F10,F0,F6 | 5 | | |
| 6. ADD.D F6, F8, F2 | 6 | 9 | 11 |

**Reservation Stations**

| | Busy | Op | Vj | Vk | Qj | Qk | A |
|---|---|---|---|---|---|---|---|
| LD1 | 0 | | | | | | |
| LD2 | 0 | | | | | | |
| AD1 | 0 | | | | | | |
| AD2 | 0 | | | | | | |
| AD3 | | | | | | | |
| ML1 | ~~1~~ 0 | MUL.D | 1.5 | 2.5 | | | |
| ML2 | 1 | DIV.D | 3.75 | 0.5 | ~~ML1~~ | | |

Cycle: 16

Register Status:

| F0 | F2 | F4 | F6 | F8 | F10 | F12 | |
|---|---|---|---|---|---|---|---|
| ~~ML1~~ | | | | | ML2 | | … |

# Detailed Example

Load: 2 cycles
Add: 2 cycles
Mult: 10 cycles
Divide: 40 cycles

Assume

R2 is 100

R3 is 200

F4 is 2.5

| Instruction | | Is | Ex | W |
|---|---|---|---|---|
| 1. L.D | F6, 34(R2) | 1 | 2 | 4 |
| 2. L.D | F2, 45(R3) | 2 | 3 | 5 |
| 3. MUL.D | F0, F2, F4 | 3 | 6 | 16 |
| 4. SUB.D | F8, F2, F6 | 4 | 6 | 8 |
| 5. DIV.D | F10,F0,F6 | 5 | 17 | |
| 6. ADD.D | F6, F8, F2 | 6 | 9 | 11 |

**Reservation Stations**

| | Busy | Op | Vj | Vk | Qj | Qk | A |
|---|---|---|---|---|---|---|---|
| LD1 | 0 | | | | | | |
| LD2 | 0 | | | | | | |
| AD1 | 0 | | | | | | |
| AD2 | 0 | | | | | | |
| AD3 | | | | | | | |
| ML1 | 0 | | | | | | |
| ML2 | 1 | DIV.D | 3.75 | 0.5 | | | |

Cycle: 17

Register Status:

| F0 | F2 | F4 | F6 | F8 | F10 | F12 | |
|---|---|---|---|---|---|---|---|
| | | | | | ML2 | | … |

# Detailed Example

Load: 2 cycles
Add: 2 cycles
Mult: 10 cycles
Divide: 40 cycles

Assume

R2 is 100

R3 is 200

F4 is 2.5

|  |  |  | Is | Ex | W |
|---|---|---|---|---|---|
| 1. | L.D | F6, 34(R2) | 1 | 2 | 4 |
| 2. | L.D | F2, 45(R3) | 2 | 3 | 5 |
| 3. | MUL.D | F0, F2, F4 | 3 | 6 | 16 |
| 4. | SUB.D | F8, F2, F6 | 4 | 6 | 8 |
| 5. | DIV.D | F10,F0,F6 | 5 | 17 |  |
| 6. | ADD.D | F6, F8, F2 | 6 | 9 | 11 |

**Reservation Stations**

|  | Busy | Op | Vj | Vk | Qj | Qk | A |
|---|---|---|---|---|---|---|---|
| LD1 | 0 |  |  |  |  |  |  |
| LD2 | 0 |  |  |  |  |  |  |
| AD1 | 0 |  |  |  |  |  |  |
| AD2 | 0 |  |  |  |  |  |  |
| AD3 |  |  |  |  |  |  |  |
| ML1 | 0 |  |  |  |  |  |  |
| ML2 | 1 | DIV.D | 3.75 | 0.5 |  |  |  |

Cycle: 18

|  | F0 | F2 | F4 | F6 | F8 | F10 | F12 |  |
|---|---|---|---|---|---|---|---|---|
| Register Status: |  |  |  |  |  | ML2 |  | … |

# Detailed Example

Load: 2 cycles
Add: 2 cycles
Mult: 10 cycles
Divide: 40 cycles

Assume

R2 is 100

R3 is 200

F4 is 2.5

| | | | Is | Ex | W |
|---|---|---|---|---|---|
| 1. | L.D | F6, 34(R2) | 1 | 2 | 4 |
| 2. | L.D | F2, 45(R3) | 2 | 3 | 5 |
| 3. | MUL.D | F0, F2, F4 | 3 | 6 | 16 |
| 4. | SUB.D | F8, F2, F6 | 4 | 6 | 8 |
| 5. | DIV.D | F10,F0,F6 | 5 | 17 | 57 |
| 6. | ADD.D | F6, F8, F2 | 6 | 9 | 11 |

**Reservation Stations**

| | Busy | Op | Vj | Vk | Qj | Qk | A |
|---|---|---|---|---|---|---|---|
| LD1 | 0 | | | | | | |
| LD2 | 0 | | | | | | |
| AD1 | 0 | | | | | | |
| AD2 | 0 | | | | | | |
| AD3 | | | | | | | |
| ML1 | 0 | | | | | | |
| ML2 | ~~1~~ 0 | DIV.D | 3.75 | 0.5 | | | |

Cycle: 57

Register Status:

| F0 | F2 | F4 | F6 | F8 | F10 | F12 | |
|---|---|---|---|---|---|---|---|
| | | | | | ~~ML2~~ | | … |

Georgia Tech College of Computing

# Timing Example

- Kind of hard to keep track with previous table-based approach
- Simplified version to track timing only

Load: 2 cycles
Add: 2 cycles
Mult: 10 cycles
Divide: 40 cycles

| Inst | Operands | Is | Exec | Wr | Comments |
|---|---|---|---|---|---|
| L.D | F6,34(R2) | 1 | 2 | 4 | |
| L.D | F2, 45(R3) | 2 | 3 | 5 | |
| MUL.D | F0,F2,F4 | 3 | 6 | 16 | |
| SUB.D | F8,F2,F6 | 4 | 6 | 8 | |
| DIV.D | F10,F0,F6 | 5 | 17 | 57 | |
| ADD.D | F6,F8,F2 | 6 | 9 | 11 | |