RISC-V Architecture & Processor Design

Week 8

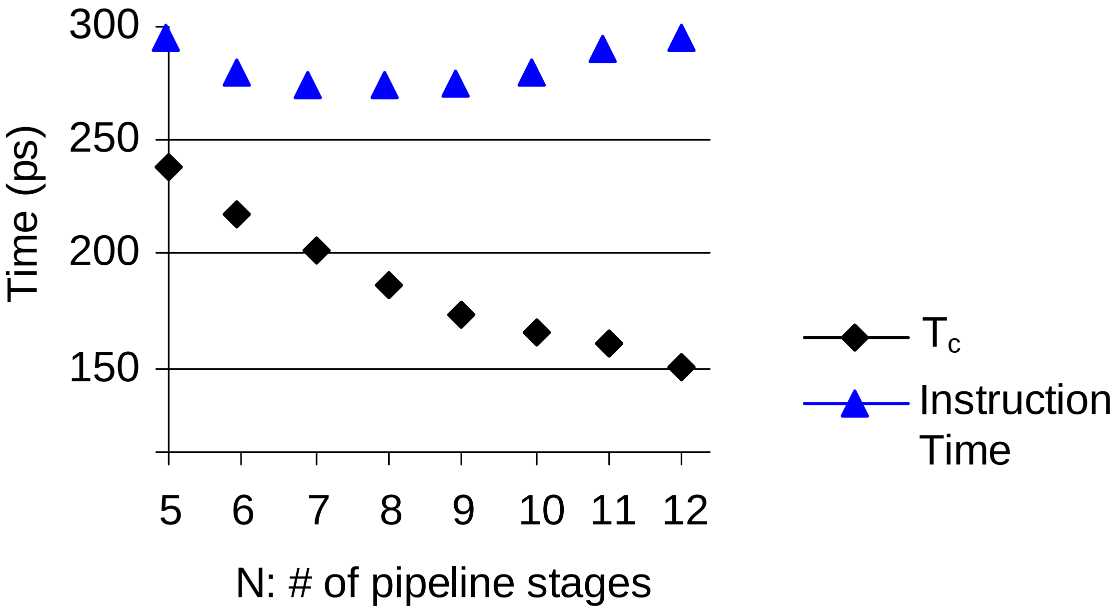
Advanced Microarchitectures: Superscalar Processors

Advanced Microarchitecture

- Deep Pipelining
- Micro-operations
- Branch Prediction
- Superscalar Processors
- Out of Order Processors
- Register Renaming
- SIMD
- Multithreading
- Multiprocessors

Deep Pipelining

- **10-20 stages typical**
- Number of stages limited by:
  - Pipeline hazards
  - Sequencing overhead
  - Power
  - Cost



Micro-operations

- Decompose complex instructions into series of simple instructions called ***micro-operations*** (*micro-ops* or *μ-ops*)
- **At run-time**, complex instructions are decoded into one or more micro-ops
- Used heavily in **CISC** (complex instruction set computer) architectures (e.g., x86)

Complex Op

```
lw s1, 0(s2), postincr 4
```

Micro-op Sequence

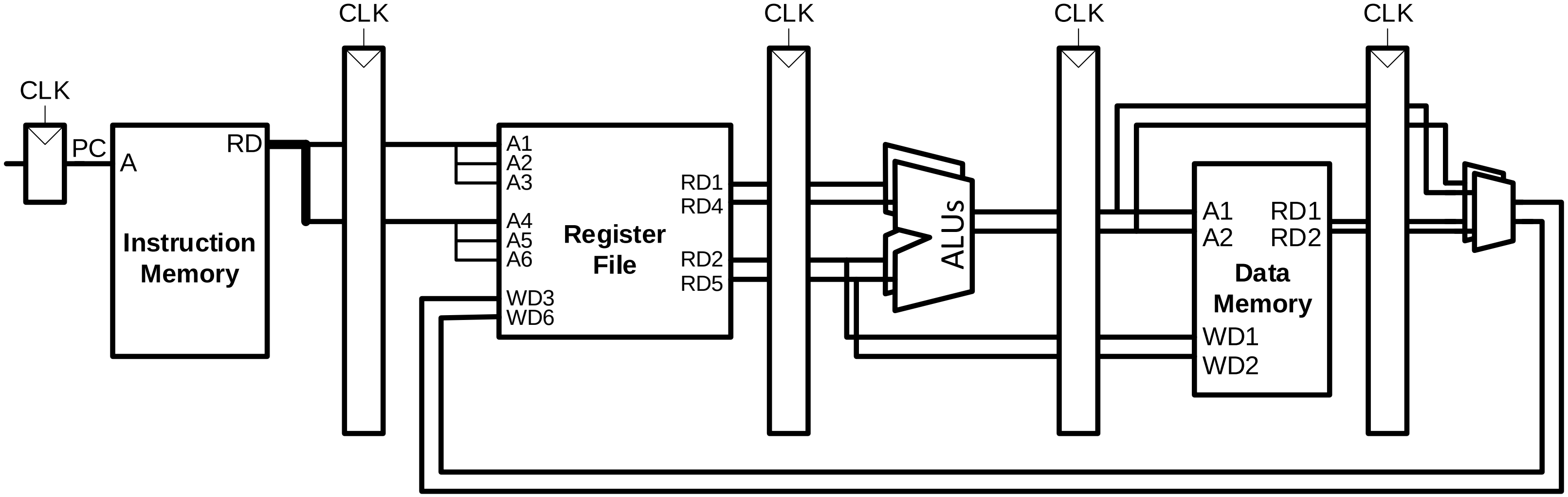
```
lw   s1, 0(s2)
addi s2, s2, 4
```

Without μ-ops, would need 2nd write port on the register file

Superscalar Processors

Superscalar Processors

- Multiple copies of datapath execute multiple instructions at once
- Dependencies make it tricky to issue multiple instructions at once

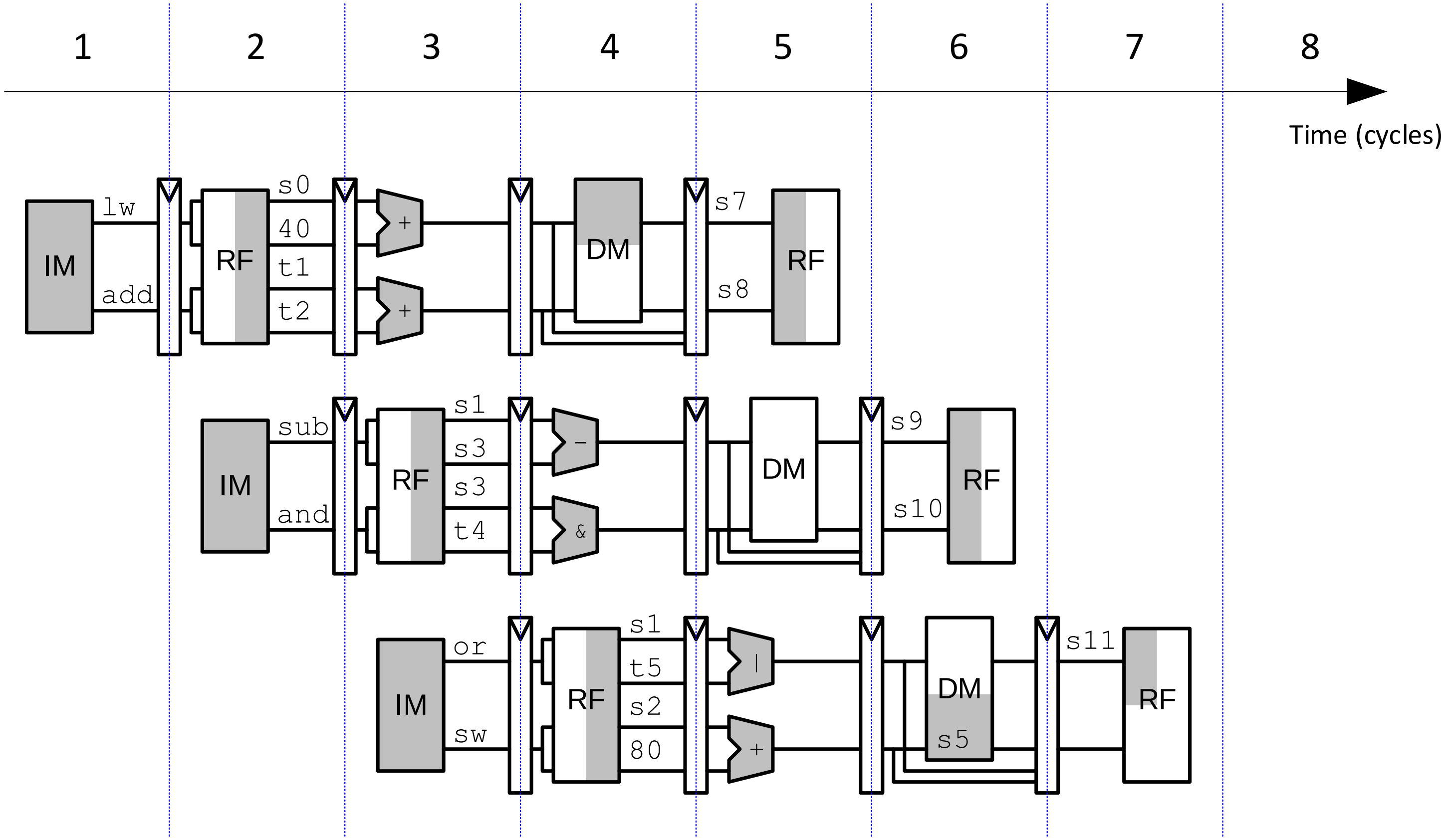


Superscalar Example

Ideal IPC: **2**

Actual IPC: **2**

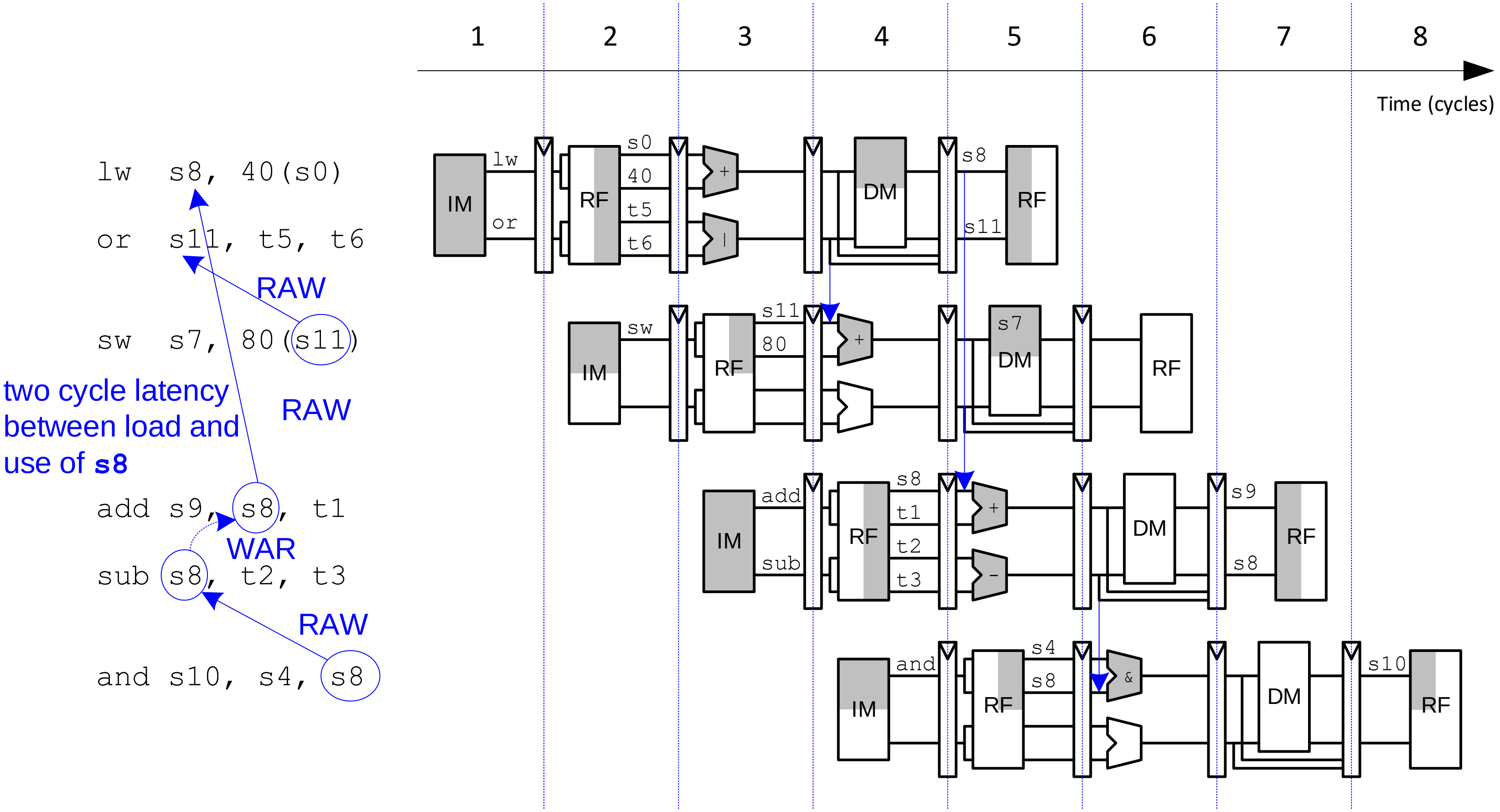
```
lw  s7, 40(s0)
add s8, t1, t2

sub s9, s1, s3
and s10, s3, t4

or  s11, s1, t5
sw  s5, 80(s2)
```



Superscalar with Dependencies

Ideal IPC: **2**

Actual IPC: **6/5 = 1.2**



```
lw   s8, 40(s0)

or   s11, t5, t6

sw   s7, 80(s11)

add  s9, s8, t1

sub  s8, t2, t3

and  s10, s4, s8
```

RAW

RAW

two cycle latency between load and use of **s8**

WAR

RAW

Out-of-Order Processors

Out of Order (OOO) Processor

- Looks ahead across multiple instructions
- Issues as many instructions as possible at once
- Issues instructions out of order (as long as no dependencies)
- **Dependencies:**
  - **RAW** (read after write): one instruction writes, later instruction reads a register
  - **WAR** (write after read): one instruction reads, later instruction writes a register
  - **WAW** (write after write): one instruction writes, later instruction writes a register

Out of Order (OOO) Processor

- **Instruction level parallelism (ILP):** number of instruction that can be issued simultaneously (average < 3)
- **Scoreboard:** table that keeps track of:
  - Instructions waiting to issue
  - Available functional units
  - Dependencies

Out of Order Processor Example

Ideal IPC: **2**

Actual IPC: **6/4 = 1.5**

```
lw   s8, 40(s0)
or   s11, t5, t6
sw   s7, 80(s11)
add  s9, s8, t1
sub  s8, t2, t3
and  s10, s4, s8
```

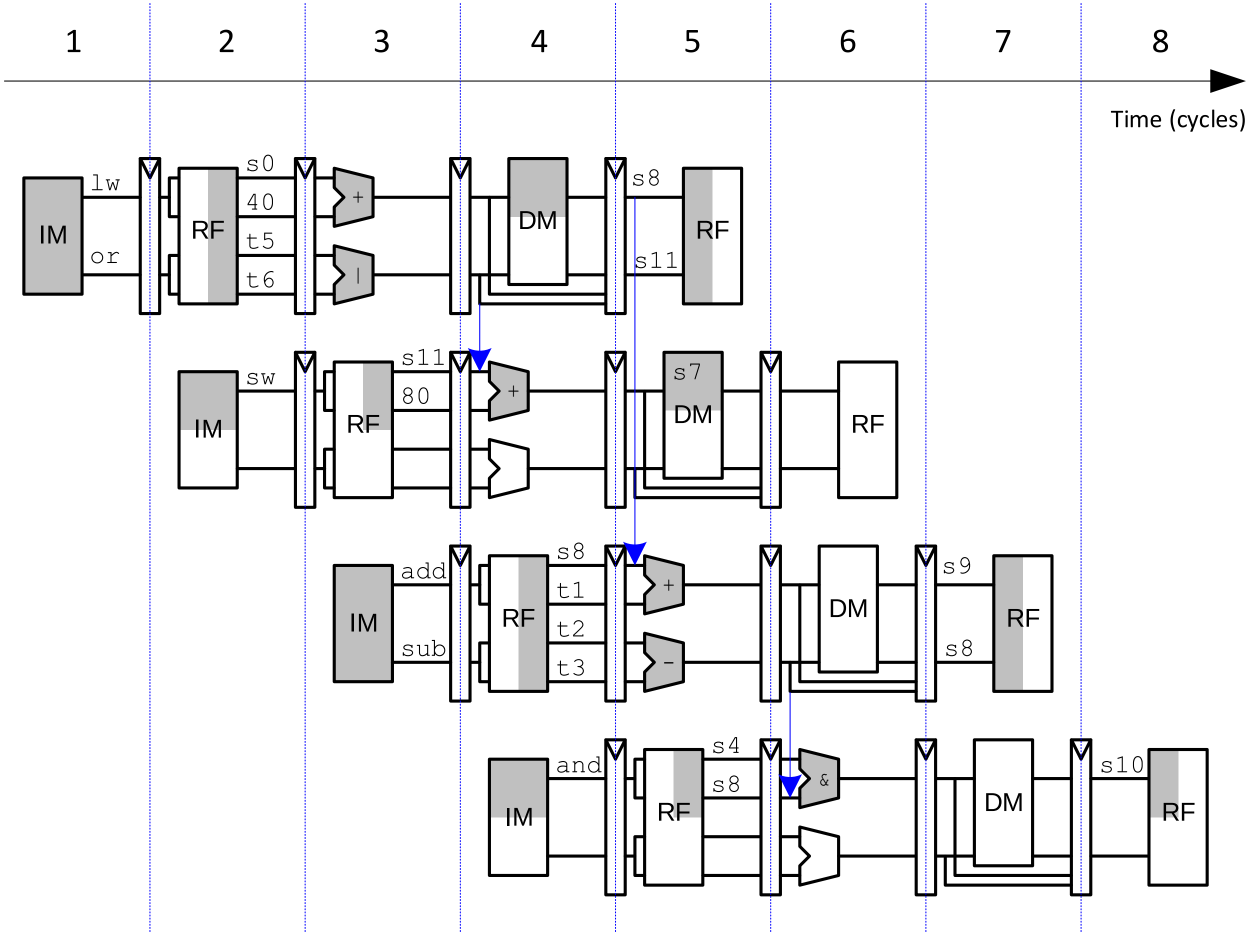
RAW

RAW

two cycle latency between load and use of R8

WAR

RAW



Register Renaming

Ideal IPC: **2**

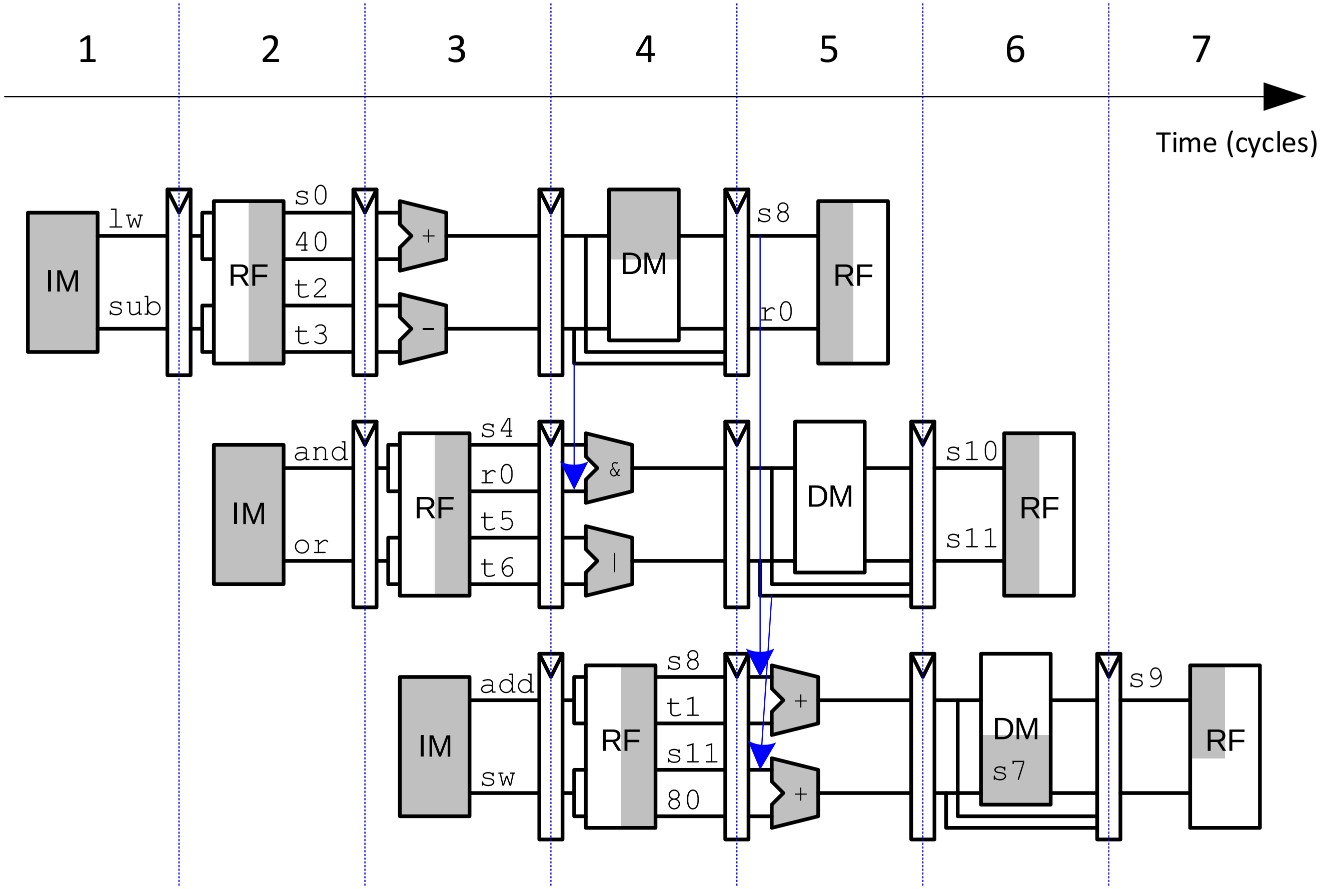
Actual IPC: **6/3 = 2**

```
lw   s8, 40(s0)

sub  r0, t2, t3

and  s10, s4, r0

or   s11, t5, t6

add  s9, s8, t1

sw   s7, 80(s11)
```

2-cycle RAW

RAW

RAW



1
2
3
4
5
6
7
Time (cycles)
IM
lw
sub
RF
s0
40
t2
t3
+
-
DM
s8
r0
RF
IM
and
or
RF
s4
r0
t5
t6
&
|
DM
s10
s11
RF
IM
add
sw
RF
s8
t1
s11
80
+
+
DM
s7
s9
RF