Datapath Control lines

Mux [RegDst
MemToReg
ALUSRC
Jump
Branch

Access [RegWrite

Memory [MemWrite
MemRead

	R-format	Load	Store	Branch	Jump
RegDst	1	0	x	x	x
MemToReg	0	1	x	x	x
ALUSRC	0	1	1	0	x
Branch	0	0	0	1	0
Jump	0	0	0	0	1
RegWrite	1	1	0	0	0
MemWrite	0	0	1	0	0
MemRead	0	1	0	0	0

Single Cycle Implementation - Performance

The clock cycle has the same length for every instruction
CPI = 1

The clock cycle is determined by the longest possible path. Let's find out what is the instruction that takes the longest time:

* The Adder for PC is not considered to take any time.
* Access times:

Memory Unit :	200 ps
ALU, Adders :	100 ps
Register file :	50 ps

R-format

fetch: Read Instruction memory ⇒ 200 ps

Decode: Read Reg file (read 2 source reg in parallel) ⇒ 50 ps

Compute: Access ALU ⇒ 100 ps

Write Back: Write into the Reg file ⇒ 50 ps

400 ps

Store

fetch: Read Instruction memory ⇒ 200 ps

Decode: Read Reg file (read 2 source reg in parallel) ⇒ 50 ps

Compute: Access ALU (compute target Address) ⇒ 100 ps

Memory Access (write data to memory) ⇒ 200 ps

550 ps

Load

fetch: Read Instruction memory ⇒ 200 ps

Decode: Read Reg file (reads only 1 source Address) ⇒ 50 ps

Compute: Access ALU (compute target Address) ⇒ 100 ps

Memory Access (read data from memory) ⇒ 200 ps

Write Back (write to Reg file) ⇒ 50 ps

600 ps

Beq

fetch: Read Instruction memory ⇒ 200 ps

Decode: Read Reg file (read 2 source reg in parallel) ⇒ 50 ps

Compute: Access ALU (check for zero)

⇆ Done in Parallel ⇒ 100 ps

Compute: Target Address (use an Adder)

350 ps

* A program with 50 instructions: 50 × length of lw (since lw is the longest instruction)