

= {
1, if opcode = "1001"
&
R[src1] = 0}

Din → Patch → Dout

ADD R0, R0, R1

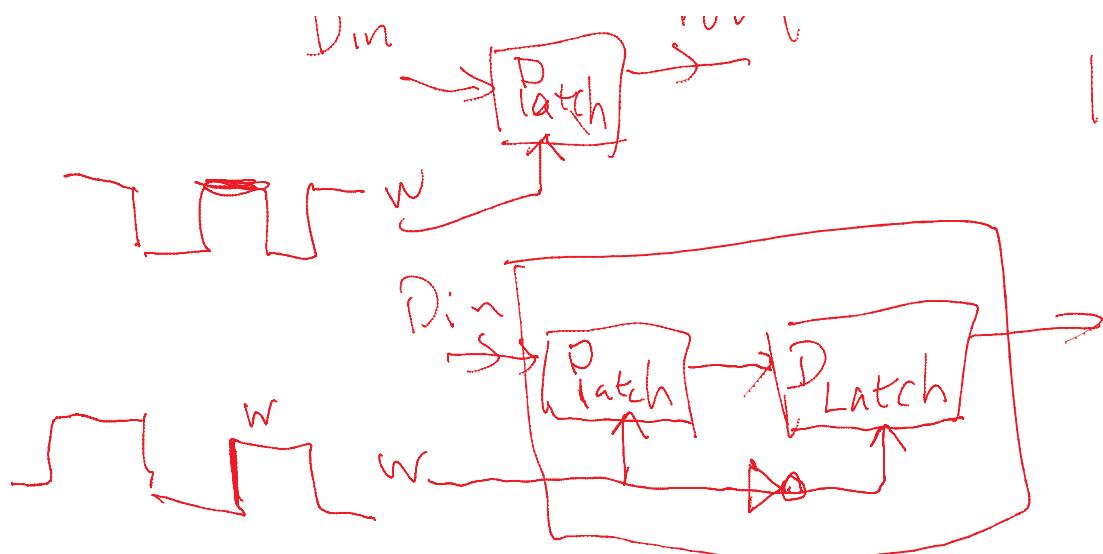
Single cycle Processor

U.S.
multiple cycle
Pipelined

CSC360
CSC372

period + ... ,

CISC372



level-triggered

half cycle for $w=1$ can
"flood" the register

edge-triggered

write can only happen
instantaneously at
the rising edge of

2nd half
cycle

How to do unbounded computation?
with finite size of program

e.g.

$$c = a \times b$$

$$\rightarrow \text{SUB } R_7, R_7, R_7$$

ADD

$$R_7, R_7, R_0$$

ADD

$$R_7, R_7, R_0 \} b \text{ times}$$

$$R_0 \leftarrow a, R_1 \leftarrow b$$

Register assignment
 $R_0 \leftarrow a, R_1 \leftarrow b$

HWK R_1, R_2, R_3 } b times R₀ ← a, R₁ ← b
R₂ ← sum, R₂ ← 1
R₃ ← 0
 sum = 0;
 while (b > 0) {
 sum = sum + a;
 b--;
 }
 return sum;

Addr SUB R₂, R₁, R₂
 0x00 Loop: ADD R₂, R₁, R₂
 04: SUB R₁, R₁, R₂
 06: BRZ R₁, Exit
 08: BRZ R₃, Loop
 Exit: - - -

We need:
flow control

Branch condition BAddr

if (condition true)
 PC ← BAddr

else ~ ~

else

$PC \leftarrow PC + 2;$

BRZ R_0 , BAddr

BRZ R_3 , Loop

1001 0 011 000010
opcode w src BAddr

address for label Loop
is 0x02

Time complexity

$O(b)$

Example : Shift-Add algorithm for Integer multiplication

Source Code

Assembly code

Machine Code

ISPRad w/ Src, SPC2 dst

```

sum = 0;
while(b > 0) {
    if(b & 0x01 == 1) {
        sum = sum + a;
        a = a << 1;
        b = b >> 1;
    }
    return sum;
}

```

Loop:

Sub R7, R7, R7	
BRZ R1, Return	
AND R4, R1, R2	
BRZ R4, Else	
ADD R7, R7, R0	

Else:

SLL R0, R0, R2	
SRL R1, R1, R2	
BRZ R3, Loop	

Return ; HALT

Machine code

Opcode	W	Src1	Src2	Dst
0001	1	111	111	111
1001	0	001		
0111	1	001	010	100
1001	0	100	001	010
...				
...				

Register Usage

R_0 : a
 R_1 : b
 R_2 : 1
 R_3 : 0

Time complexity

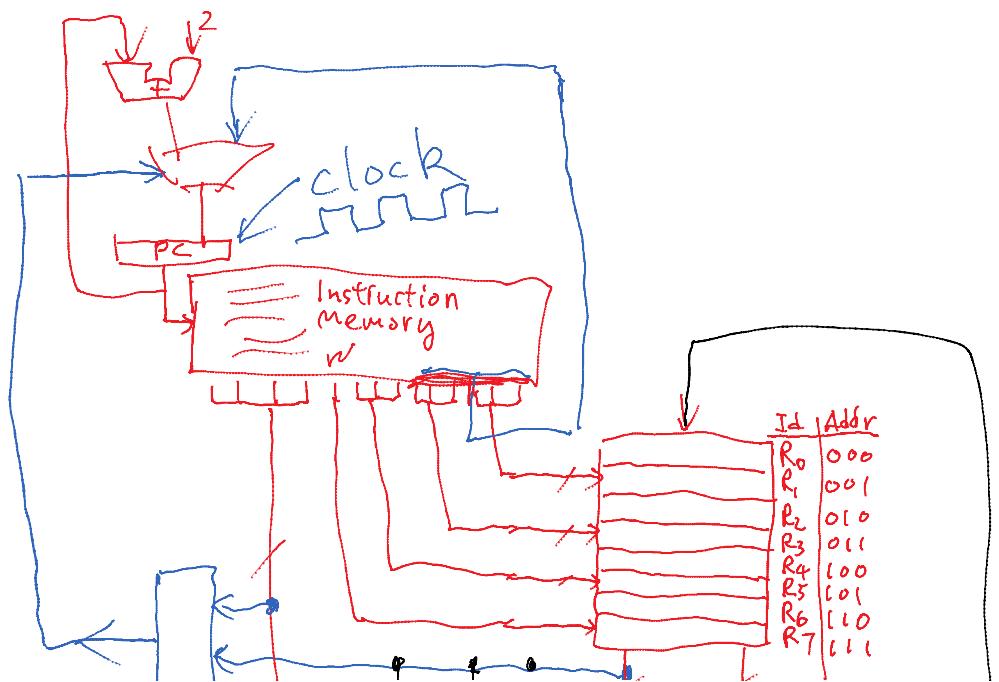
$O(\log_2 b)$

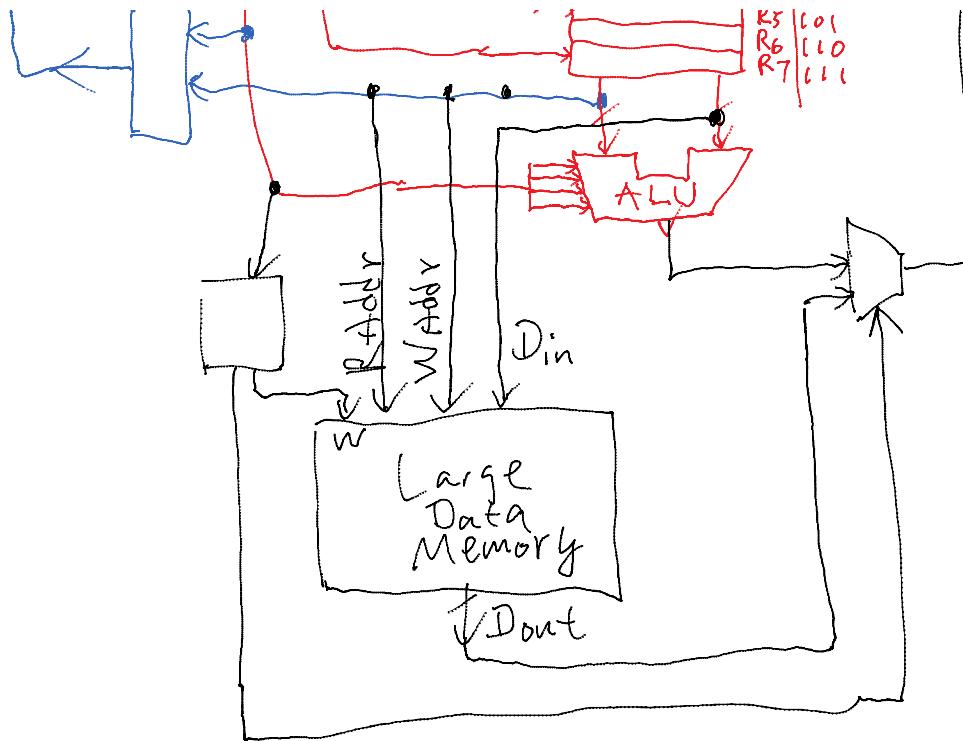
$R_3 := 0$
 $R_4 :=$

$O(\log_2 b)$

$R_7 := \text{sum}$

Add a large data memory





Two new instructions:

LD src1, dst

$$R[dst] \leftarrow M[R[src1]]$$

ST src1, src2

$M[R[\underline{\text{src1}}]] \leftarrow R[\underline{\text{src2}}]$