①

Cycle	State	LD.PC	LD.MAR	LD.MDR	LD.REG	LD.CC	GateALU	GatePC	ALUK	PCMUX	[illegible]	[illegible]
1	18	1	1	0	0	0	0	1	0	00		
2	33	0	1	1	0	0	0	0	X	X		
3	35	0	0	1	0	0	0	0	X	X		
4	32	0	0	0	0	0	0	0	X	X		
5	5	0	0	0	1	1	1	0	01	X		
6												
7												
8												
9												
10												

②

- An addressing mode determines how a machine interprets operand bits

  (think of it like the parameter data types of a high level function)

- Register, Immediate, PC-relative, Base + Offset, Indirect

- x3F20 : ST operation
  - Register
  - PC-relative

- x14B5 : ADD operation
  - Register
  - Immediate (not always)

③

- LD: 2 times,
  - Read instruction from memory into IR
  - Read from designated location

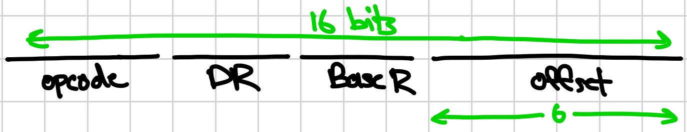
- STR: 2 times,
  - Read instruction from memory into IR
  - Write to designated location

- LDI: 3 times,
  - Read instruction from memory into IR
  - Read address from PC-relative location
  - Read from address location

- BR: 1 time,
  - Read instruction from memory into IR

4

- LDR:



$$[-2^5, 2^5-1] = [-32, 31] = [-10\underbrace{0000}_{-x20}, 01\underbrace{1111}_{x1F}]$$

$$\begin{array}{r} x6880 \\ -\ x0020 \\ \hline x6860 \end{array} \qquad \begin{array}{r} x6880 \\ +x001F \\ \hline 689F \end{array}$$

$[x6860, x689F]$

- $[0, 2^6-1] = [0, 63] = [0, \underbrace{11}_{x3}\underbrace{1111}_{F}]$

$$\begin{array}{r} x6880 \\ +\ x003F \\ \hline x68BF \end{array}$$

$[x6880, x68BF]$

- 9-bit Offset

$$[-2^8, 2^8-1] = [-256, 255] = [\underset{-x100}{-100000000}, \underset{xFF}{11111111}]$$

$$\begin{array}{r} x4951 \\ -x0100 \\ \hline x4851 \end{array} \qquad \begin{array}{r} x4\overset{1}{9}\overset{1}{5}1 \\ +x00FF \\ \hline x4A50 \end{array}$$

$[x4851, x4A50]$

⑤

- $\lceil \log_2 250 \rceil = 8$
- $\lceil \log_2 100 \rceil = 7$
- $32 - 8 - 7 - 7 - 7 = 3$

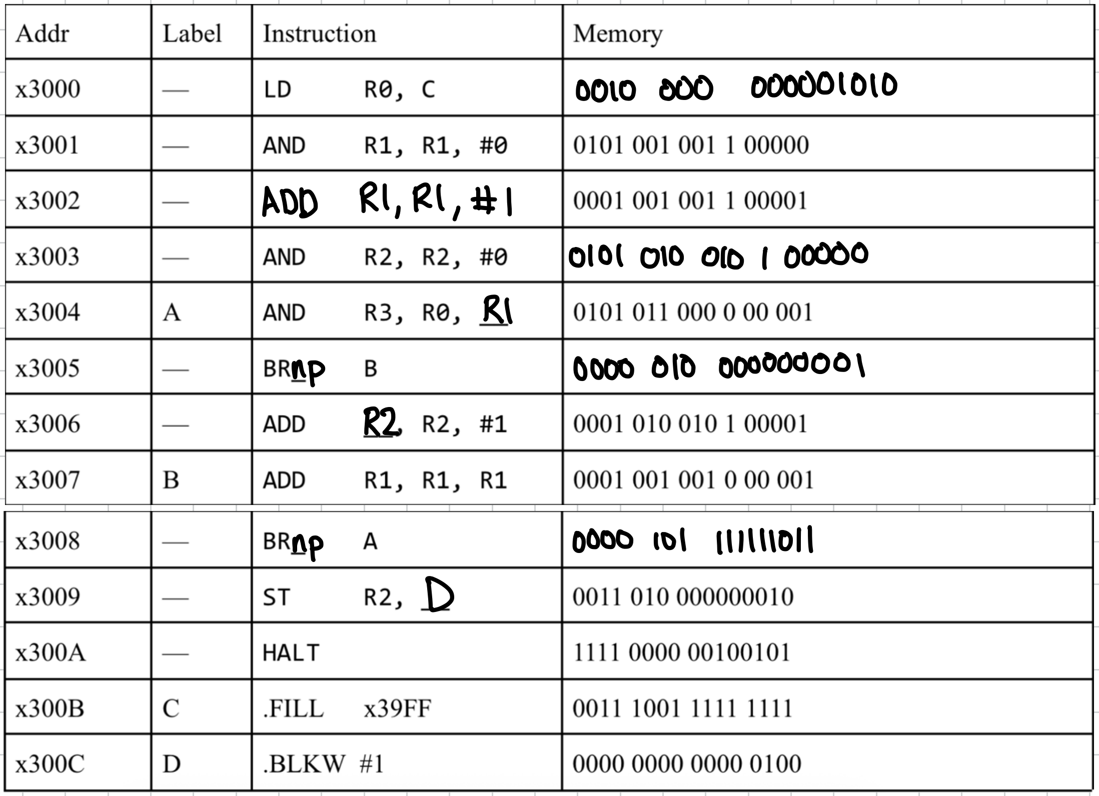
⑥

	PC	IR	MAR	MDR	R0	R1
Fetch	x3003	x6200	x3002	x6200	x3001	x3429
Decode	x3003	x6200	x3002	x6200	x3001	x3429
Evaluate Address	x3003	x6200	x3001	x6200	x3001	x3429
Fetch Operands	x3003	x6200	x3001	x2001	x3001	x3429
Execute	x3003	x6200	x3001	x2001	x3001	x3429
Store Result	x3003	x6200	x3001	x2001	x3001	x2001

0110 0010 0000 0000

LDR R1, R0, #0

7



Addr	Label	Instruction	Memory
x3000	—	LD R0, C	0010 000 000001010
x3001	—	AND R1, R1, #0	0101 001 001 1 00000
x3002	—	ADD R1, R1, #1	0001 001 001 1 00001
x3003	—	AND R2, R2, #0	0101 010 010 1 00000
x3004	A	AND R3, R0, R1	0101 011 000 0 00 001
x3005	—	BRnp B	0000 010 000000001
x3006	—	ADD R2, R2, #1	0001 010 010 1 00001
x3007	B	ADD R1, R1, R1	0001 001 001 0 00 001
x3008	—	BRnp A	0000 101 111111011
x3009	—	ST R2, D	0011 010 000000010
x300A	—	HALT	1111 0000 00100101
x300B	C	.FILL x39FF	0011 1001 1111 1111
x300C	D	.BLKW #1	0000 0000 0000 0100

- Counts the number of 0-bit's @ x300B and stores result @ x300C

8

	PC	MAR	MDR	IR	R0	R1
Before 1st instruction	x3000	x0000	-x0000	-x0000	x0000	x0000
After 1st instruction	x3001	x3005	x8667	x2004	x8667	x0000
After 2nd instruction	x3002	x3001	x5021	x5021	x0001	x0000
After 3rd instruction	x3003	x3002	x0401	x0401	x0001	x0000
After 4th instruction	x3004	x3003	x1261	x1261	x0001	x0001

x5021 ⇒ 0101 0000 0010 0001
AND R0, R0, #1

x0401 ⇒ 0000 0100 0000 0001
BRz #1

ADD R1, R1, #1 ⇒ 0001 001 001 1 00001
1 2 6 1

LD R0, x04

0010 000 000000100

- This program checks if the LSB in x3005 is a 1 (odd), and sets R1 to x0001 for odd and x0000 for even.