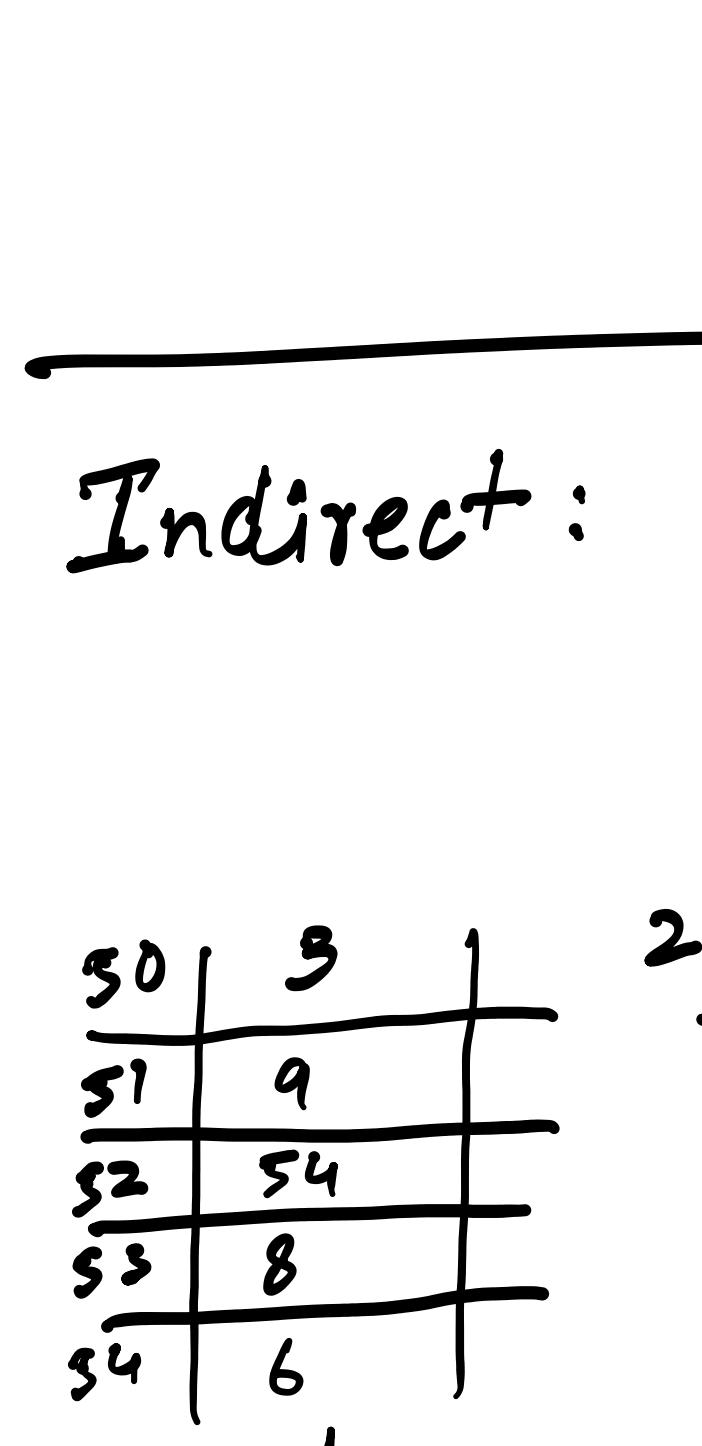


Direct Addressing:

Mnemonic

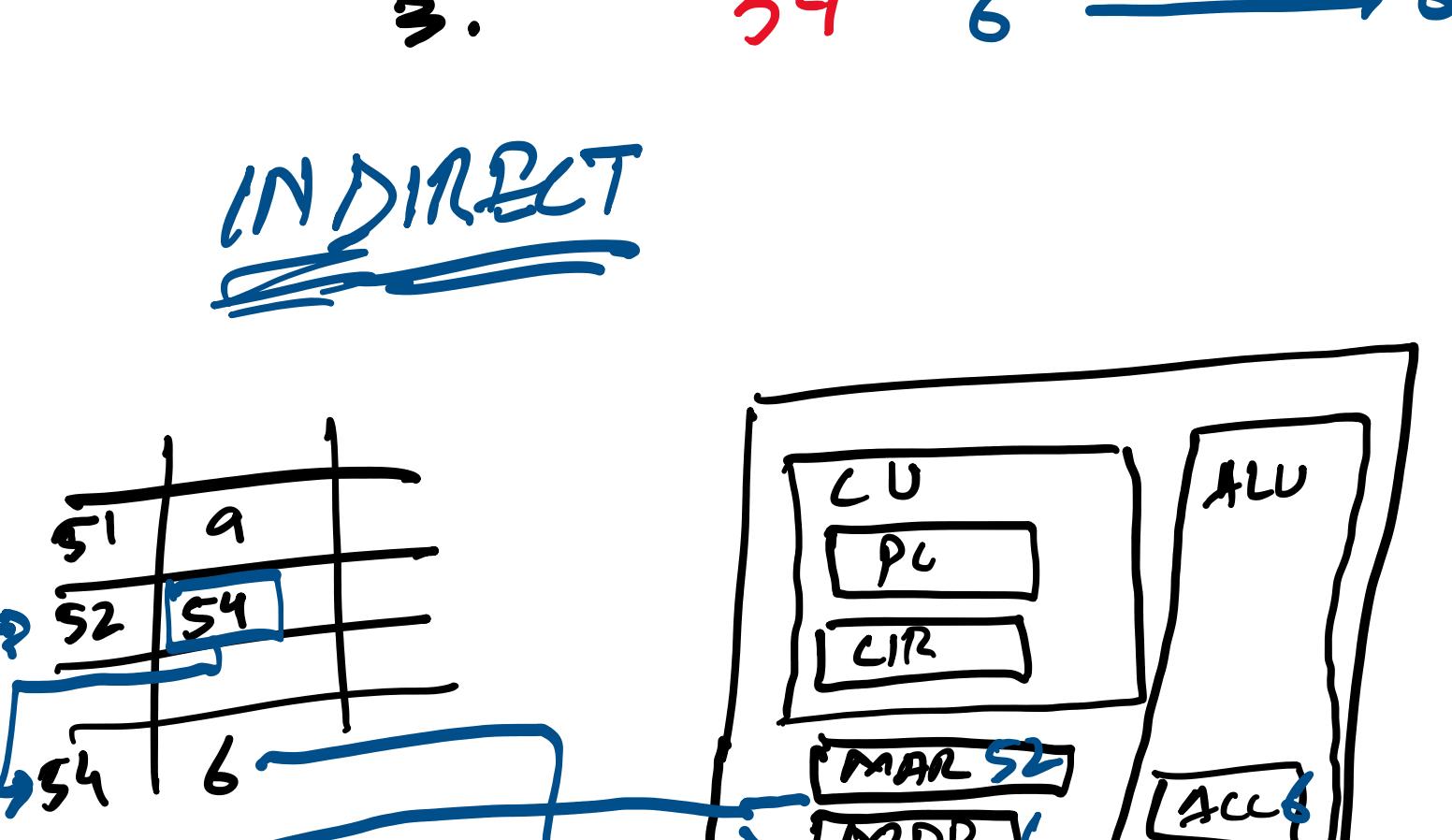
LDD	<addr>
Op-Code	operand



200 | LDD 51

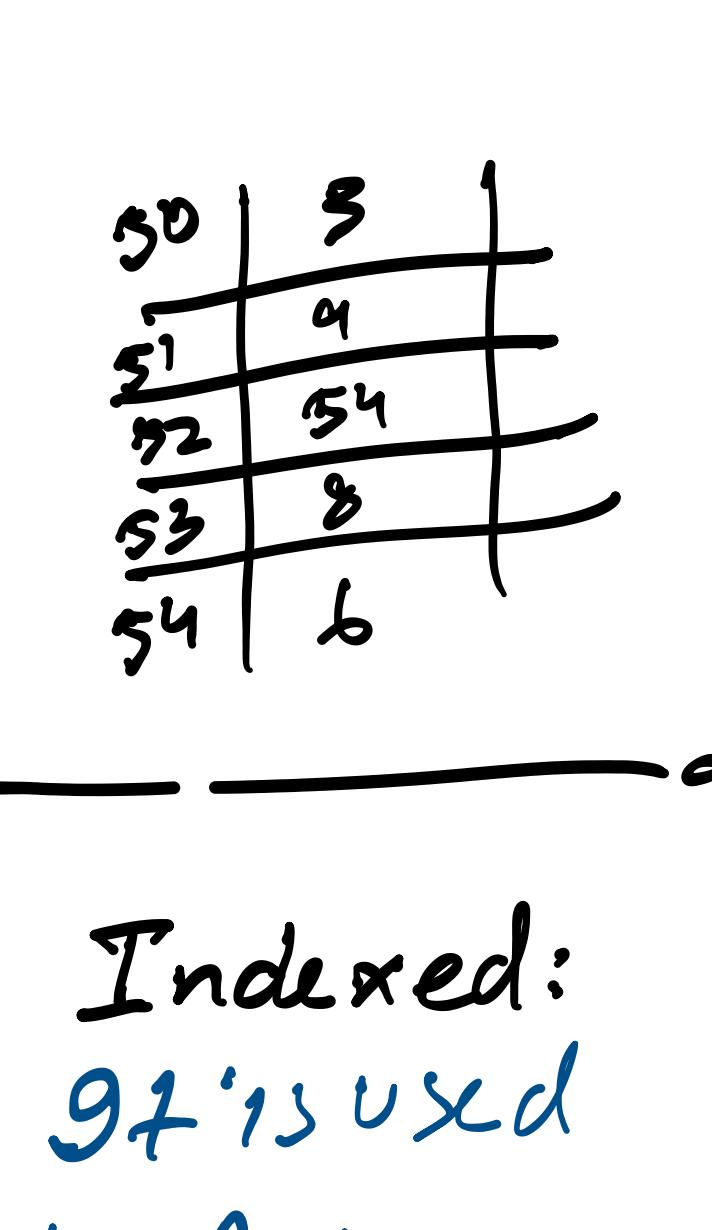
Clock Cycles
1. #9C MAR MDR CIR ACC
2. 200 LDD 51 9

DIRECT



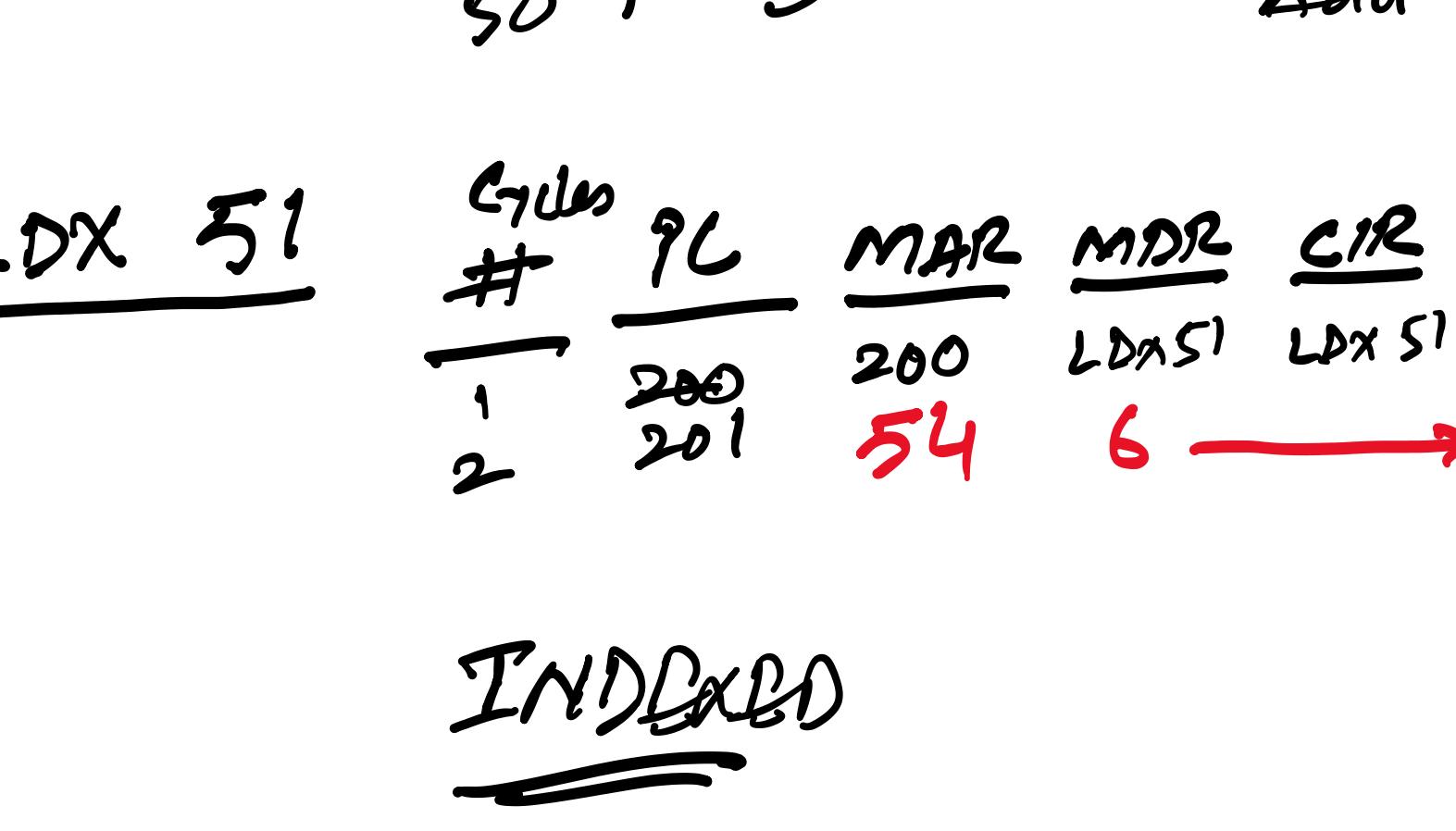
Indirect:

LDI <addr>



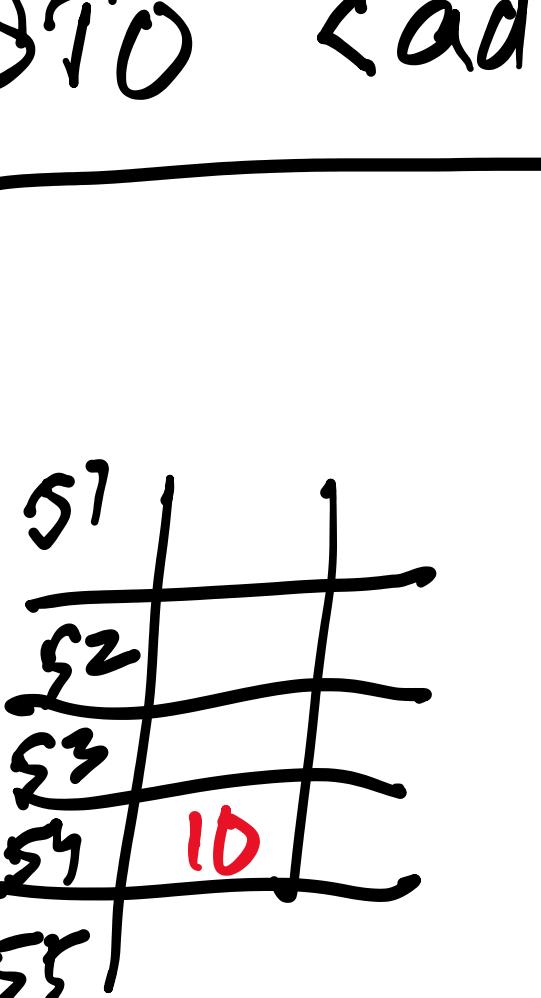
200 | LDI 52 #9C MAR MDR CIR ACC
1. 200 200 LDI 52 LDI 52
2. 201 52 54
3. 54 6 → 6

INDIRECT

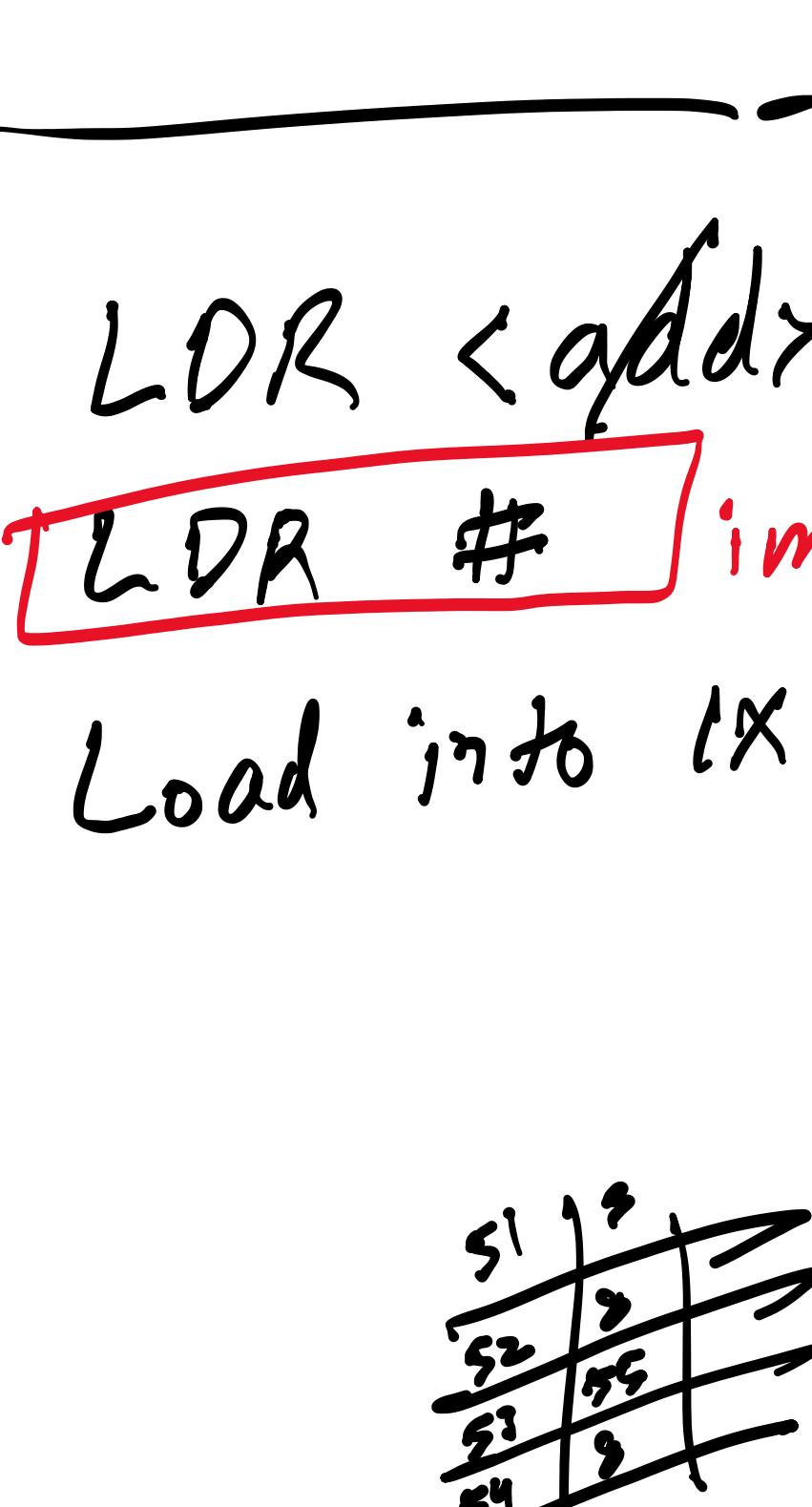


IMMEDIATE:

LDM #



200 | LDM #4 #PC MAR MDR CIR ACC
1. 200 200 LDM #4 LDM #4
2. 201 4



LDX <addr>

makes ux of index (1x) register.

E.g. LDX 50 1x
Base Add Offset
50 + 1x = 53 Actual Add.



STO <addr>

200 | STO 54

Clock Cycles
#PC MAR MDR CIR STO 54
1. 200 200 STO 54 54
2. 201 10 10

ACC 10

- Bits at 1 memory location
- Width of System Bus
- Size of registers

WORD Size Amount of bits a computer moves, stores, processes and understands in one go.

MAR = 32 bits Add. Bus 32
 $2^{32} = 4\text{GB}$ ✓ 11111111111111111111111111111111
8 GB RAM X

Instruction is 32 bit word size.

32 bits instruction
OP-CODE/Mem. 5 bits Operand 12 bit immediate.

LDD $2^2 = 4096$ (4K)
LDI 597 within 2¹² range

Main Memory 4096 bytes 4KB

LDI opcode is used to access addresses outside the range of available operand width.

64 Bits word size

64 bits
40 5:3 24

$2^{24} = 17\text{ million.}$