Amrit Singh

Dr. Weide Chang

CSC 137

10/2/18

5-1

1. Indirect: 1bit, Address: 18bit, Reg: 6 bits, OP-code: 7bits.

5-2

The difference between Direct and Indirect address instruction is that Direct is a one memory reference, it gets the operand from the given address. Indirect has two memory references and gets the address of the operand from the given address and uses this new address to get the value of the operand.

5-3

1. Connect memory to BUS, Read from memory, load from BUS to IR. IR <- M[AR]
2. Connect register TR to the BUS, load from bus to PC, PC <-TR
3. AC bus write to memory and load to DR. M[AR] <- AC, DR <-AC
4. ADD DR or INPR to AC. AC <-AC+DR

5-4

a). AR <- PC

S1S2S3: 0 1 0(PC)   LD of register: AR

*b)*. IR <- M[AR]

S1S2S3: 1 1 1(M)   LD of register:  IR                  Memory :Read

*c)*. M[AR] <- TR

S1S2S3: 1 1 0(TR) Memory: White

*d)*. AC <- DR, DR<-AC

S1S2S3: 1 0 0(AC)    LD of register : AC, DR        Adder: Transfer DR to AC

5-5

A. IR <- M[PC] The program counter (PC) must be moved to address register(AR) as PC cannot provide address to memory.

B. AC <- AC + TR add operations are done with AC & DR, TR must be transferred to DR first.

C. DR <- DR + AC (AC doesn’t change) add operations should be stored in AC not DR.

5-6

a)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0001 | 0000 | 0010 | 0100 |  |
| Hex = | 1 | 0 | 2 | 4 |

ADD M[024] to AC

b)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1011 | 0001 | 0010 | 0100 |  |
| Hex = | B | 1 | 2 | 4 |

Stores AC in M[M[124]]

c)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | 0111 | 0000 | 0010 | 0000 | |  | | |
| Hex = 7 0 2 0 |  |  |  | |  |

Increments AC

5-7

The two instructions needed are the CLE Clear E and

CME Complement E instructions

5-9

AC = Accumulator; IR = Instruction Register;

AR = Address Register;  PC = Program Counter;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **E** | **AC** | **PC** | **AR** | **IR** |
| **Initial** | 1 | A937 | 021 |  |  |
| **CLA** | 1 | 0000 | 022 | 800 | 7800 |
| **CLE** | 0 | A937 | 022 | 400 | 7400 |
| **CMA** | 1 | 56C8 | 022 | 200 | 7200 |
| **CME** | 0 | A937 | 021 | 100 | 7100 |
| **CIR** | 1 | D49B | 022 | 080 | 7080 |
| **CIL** | 1 | 526F | 022 | 040 | 7040 |
| **INC** | 1 | A938 | 022 | 020 | 7020 |
| **SPA** | 1 | A937 | 022 | 010 | 7010 |
| **SNA** | 1 | A937 | 023 | 008 | 7008 |
| **SZA** | 1 | A937 | 022 | 004 | 7004 |
| **SZE** | 1 | A937 | 022 | 002 | 7002 |
| **HLT** | 1 | A937 | 022 | 001 | 7001 |

5-10

AC = Accumulator; AR = Address Register; PC = Program Counter;

DR = Data Register; IR = Instruction Register;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **PC** | **AR** | **DR** | **AC** | **IR** |
| **Initial** | 021 |  |  | A937 |  |
| **AND** | 022 | 083 | B8F2 | A832 | 0083 |
| **ADD** | 022 | 083 | B8F2 | 6229 | 1083 |
| **LDA** | 022 | 083 | B8F2 | B8F2 | 2083 |
| **STA** | 022 | 083 |  | A937 | 3083 |
| **BUN** | 083 | 083 |  | A937 | 4083 |
| **BSA** | 084 | 084 | = | A937 | 5083 |
| **ISZ** | 022 | 083 | B8F3 | A937 | 6083 |

5-11

AR = Address Register; SC = Sequence Counter; PC = Program Counter;

DR = Data Register; IR = Instruction Register;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **PC** | **AR** | **DR** | **AC** | **SC** |
| **Initial** | 7FF |  |  |  | 0 |
| **T0** | 7FF | 7FF |  |  | 1 |
| **T1** | 800 | 7FF |  | EA9F | 2 |
| **T2** | 800 | A9F |  | EA9F | 3 |
| **T3** | 800 | C35 |  | EA9F | 4 |
| **T4** | 800 | C35 | FFFF | EA9F | 5 |
| **T5** | 800 | C35 | 0000 | EA9F | 6 |
| **T6** | 801 | C35 | 0000 | EA9F | 0 |

5-12

a). 3AF | 932E         9 = 1001 I = 1 001 = ADD, so next is to ADD 32E

    32E | 09AC

    9AC | 8B9F

b). 32E | 09AC        AC = 7EC3   7EC3 + 8B9F = 0A62

    9AC | 8B9F         DR = 8B9F