**SUBMITTED BY:- ABHIMANYU**

**ROLL NO:- 20191402**

**QUESTION-3:**

**Write an assembly language program to simulate ADD operation on two user-entered numbers.**

**SOLUTION:**

INP

STA A

INP

ADD A

OUT

HLT

A**:** **.**data 2 0

**QUESTION-4:**

**Write an assembly language program to simulate SUBTRACT operation on two user-entered numbers.**

**SOLUTION:**

**INP**

**STA A**

**INP**

**STA B**

**L1: SZA**

**BUN L2**

**LDA A**

**OUT**

**HLT**

**L2: LDA A**

**ADD C**

**STA A**

**LDA B**

**ADD C**

**STA B**

**BUN L1**

**A: .data 2 0**

**B: .data 2 0**

**C: .data 2 -1**

**QUESTION-5:**

**Write an assembly program to simulate the following logical operations on two user-entered numbers.**

1. **AND**
2. **OR**
3. **NOT**
4. **XOR**
5. **NOR**
6. **NAND**

**SOLUTION:**

1. INP

STA A

INP

AND A

OUT

HLT

A**:** **.**data 2 0

1. INP

STA A

INP

ADD A

OUT

HLT

A**:** **.**data 2 0

1. INP

CMA

STA A

INP

CMA

OUT

LDA A

OUT

HLT

A**:** **.**data 2 0

1. INP

STA A

CMA

STA CA

INP

STA B

CMA

STA CB

AND A

STA R

LDA CA

AND B

ADD R

OUT

HLT

A**:** **.**data 2 0

CA**:** **.**data 2 0

B**:** **.**data 2 0

CB**:** **.**data 2 0

R**:** **.**data 2 0

1. INP

STA A

INP

ADD A

CMA

OUT

HLT

A**:** **.**data 2 0

1. INP

STA A

INP

AND A

CMA

OUT

HLT

A**:** **.**data 2 0

**QUESTION-6:**

**Write an assembly language program to simulate MULTIPLY operation on two user-entered numbers.**

**SOLUTION:**

INP

STA A

INP

STA B

L1: SZA

BUN L2

LDA RES

OUT

HLT

L2: LDA RES

ADD A

STA RES

LDA B

ADD C

STA B

BUN L1

A**:** **.**data 2 0

B**:** **.**data 2 0

C**:** **.**data 2 -1

RES**:** **.**data 2 0

**QUESTION-7:**

**Write an assembly program to for simulating following memory-reference instructions.**

1. **ADD**
2. **LDA**
3. **STA**
4. **BUN**
5. **ISZ**

**SOLUTION:**

LDA A

L1: OUT

ADD C

STA A

ISZ B

BUN L1

HLT

A: .data 1 [10]

B: .data 1 [-10]

C: .data 1 [-1]

**QUESTION-8:**

**Write an assembly language program to simulate the machine for the following register reference instructions and determine the content of AC, E, PC, AR and IR registers in decimal after the execution.**

1. **CLE**
2. **CMA**
3. **CME**
4. **HLT**

**SOLUTION:**

Let us assume that initially**:-**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **AC** | **E** | **PC** | **AR** | **IR** |
| A937 | 1 | 022 | 000 | 0000 |

1. CLA

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **AC** | **E** | **PC** | **AR** | **IR** |
| 0000 | 1 | 023 | 022 | E800 |

1. CMA

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **AC** | **E** | **PC** | **AR** | **IR** |
| 56C8 | 1 | 023 | 022 | E400 |

1. CME

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **AC** | **E** | **PC** | **AR** | **IR** |
| A937 | 0 | 023 | 022 | E100 |

1. HLT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **AC** | **E** | **PC** | **AR** | **IR** |
| A937 | 1 | 023 | 022 | E001 |

**QUESTION-9:**

**Write an assembly language program to simulate the machine for the following register reference instructions and determine the content of AC, E, PC, AR and IR registers in decimal after the execution.**

1. **INC**
2. **SPA**
3. **SNA**
4. **SZE**

**SOLUTION:**

Let us assume that initially**:-**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **AC** | **E** | **PC** | **AR** | **IR** |
| A937 | 1 | 022 | 000 | 0000 |

1. INC

**AFTER:-**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **AC** | **E** | **PC** | **AR** | **IR** |
| A938 | 1 | 023 | 022 | E020 |

1. SPA

**AFTER:-**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **AC** | **E** | **PC** | **AR** | **IR** |
| A937 | 1 | 023 | 022 | E010 |

1. SNA

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **AC** | **E** | **PC** | **AR** | **IR** |
| A938 | 1 | 023 | 022 | E008 |

1. SZE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **AC** | **E** | **PC** | **AR** | **IR** |
| A938 | 1 | 023 | 022 | E002 |

**QUESTION-10:**

**Write an assembly language program to simulate the machine for the following register reference instructions and determine the content of AC, E, PC, AR and IR registers in decimal after the execution.**

1. **CIR**
2. **CIL**

**SOLUTION:**

Let us assume that initially**:-**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **AC** | **E** | **PC** | **AR** | **IR** |
| A937 | 1 | 022 | 000 | 0000 |

1. CIR

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **AC** | **E** | **PC** | **AR** | **IR** |
| D26E | 1 | 023 | 022 | E080 |

1. CIL

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **AC** | **E** | **PC** | **AR** | **IR** |
| 549B | 1 | 023 | 022 | E040 |

**QUESTION-11:**

**Write an assembly program that reads in integer and adds them together; until a negative non-zero number is read in. Then it outputs the sum (not including the last number).**

**SOLUTION:**

L1: INP

STA A

SNA

BUN L2

LDA SUM

OUT

HLT

L2: LDA SUM

ADD A

STA SUM

BUN L1

A: .data 2 0

SUM: .data 2 0

**QUESTION-12:**

**Write an assembly program that reads in integer and adds them together; until zero is read in. Then it outputs the sum.**

**SOLUTION:**

L1: INP

STA A

SZA

BUN L2

LDA SUM

OUT

HLT

L2: LDA SUM

ADD A

STA SUM

BUN L1

A: .data 2 0

SUM: .data 2 0

**QUESTION-13:**

**Create a machine for the following instruction format:**

**15 13 12 11 0**

|  |  |  |
| --- | --- | --- |
| **Opcode** | **I** | **Address** |

**The instruction format contains a 3-bit opcode, a 1-bit addressing mode and a 12-bit address. Write an assembly program to simulate the machine for addition of two numbers with I=0(Direct Address) and address part=082. The instruction to be stored at address 022 in RAM, initialize the memory work with any decimal value at 082. Determine the contents of AC, DR, PC, AR and IR in decimal after the execution.**

**SOLUTION:**

Let us consider the initial content of registers:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **AC** | **DR** | **PC** | **AR** | **IR** |
| 5 | 0 | 34 | 0 | 0 |

and memory word at 082 address part is: 10 (in decimal)

Program:-

ADD 0 130

Result:-

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **AC** | **DR** | **PC** | **AR** | **IR** |
| 15 | 10 | 35 | 130 | 8322 |

**QUESTION-14:**

**Simulate the machine for the memory-reference instructions referred in above question with I =1 (Indirect Address) and address part = 082. The instruction to be stored at address 026 in RAM . Initialize the memory word at address 082 with the value 298. Initialize the memory word at address 298 with operand 632 and AC with 937. Determine the contents of AC, DR, PC, AR and IR in decimal after the execution.**

**SOLUTION:**

Program:-

ADD 1 130

Result:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **AC** | **DR** | **PC** | **AR** | **IR** |
| 1569 | 632 | 39 | 298 | 12418 |

**QUESTION-15:**

**The instruction format contains 3 bits of opcode, 12 bits for address and 1 bit for addressing mode. There are only two addressing modes, I = 0 is direct addressing and I = 1 is indirect addressing. Write an assembly program to check the I bit to determine the addressing mode and jump accordingly.**

**SOLUTION:**

Let us consider address part = 082 and the instruction is stored at address 022 in RAM. Memory word at address 082 contains 255 and memory word at address 255 contains 550 and AC is 750.

1. **For Direct Addressing :-**

Program:

ADD 0 130

Result:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **AC** | **DR** | **PC** | **AR** | **IR** |
| 1005 | 255 | 35 | 130 | 8322 |

1. **For Indirect Addressing :-**

Program:

ADD 1 130

Result:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **AC** | **DR** | **PC** | **AR** | **IR** |
| 1300 | 550 | 35 | 255 | 12418 |