COA PROJECT : ISA-CPU DESIGN PART 1

DATE : 13-10-2020

TEAM MEMBERS

19ucc002- Madhur verma

19ucc023 – Mohit Akhouri

19ucc026 – Divyansh Rastogi

19ucc043 – Aditya Pandey

ISA DESIGN

*Constraints :*

1 ) Word length = 8 bits ( 1 byte )

2 ) Memory address length = 8 bits ( 1 byte )

3 ) Size of register = 8 bits ( 1 byte )

*The Design of the ISA is as follows :*

( 1 ) General purpose CPU registers :

Total number of registers taken = 8

The representation of various registers are :

|  |  |
| --- | --- |
| **Registers pnemonics** | **Representation** |
| R0 | 000 |
| R1 | 001 |
| R2 | 010 |
| R3 | 011 |
| R4 | 100 |
| R5 | 101 |
| R6 | 110 |
| R7 | 111 |

( 2 ) Instruction set design :

Size of opcode = 5 bits

Total size of various instructions = 16 bits

*Different Types of instructions and their representations :*

|  |  |
| --- | --- |
| **Instruction pnemonics** | **Instruction representation** |
| LOAD | 00000 |
| COPY | 00001 |
| AND | 00010 |
| OR | 00011 |
| NOT | 00100 |
| XOR | 00101 |
| XNOR | 00110 |
| NAND | 00111 |
| NOR | 01000 |
| ADD | 01001 |
| SUB | 01010 |
| MULT | 01011 |
| DIV | 01100 |
| JUMP | 01101 |
| JMPEQ | 01110 |
| JMPLT | 01111 |
| JMPGT | 10000 |

*Various Instruction Formats :*

1 ) For Load instructions , logical NOT instruction and Conditional branch instructions :

|  |  |  |
| --- | --- | --- |
| Opcode ( 5 bits ) | Immediate value ( 3 bits ) | Destination operand ( 8 bits ) |

Where destination operand is memory operand

For Ex :

LOAD #7, 3

00000 111 00000011

|  |  |  |
| --- | --- | --- |
| Opcode ( 5 bits ) | Immediate value ( 8 bits ) | Destination operand ( 3 bits ) |

For Ex :

LOAD #7, R4

00000 00000111 100

NOT #6 , R4

00100 00000110 100

|  |  |  |
| --- | --- | --- |
| Opcode ( 5 bits ) | Source Operand ( 8 bits ) | Destination operand ( 3 bits ) |

Where source operand = memory operand and destination operand = register operand

For Ex :

LOAD 3 , R4

00000 00000011 100

|  |  |  |
| --- | --- | --- |
| Opcode ( 5 bits ) | Source operand ( 3 bits ) | Destination operand ( 8 bits ) |

Where source operand = register operand and destination operand = memory operand

For Ex :

LOAD R4 , 3

00000 100 00000011

JUMPEQ R2 , 1

01110 010 00000001

2 ) For Logical instructions ( AND , OR , XOR , NAND , NOR , XNOR ) and

Arithmetic instructions ( ADD , MULT , DIV and SUB ) :

|  |  |  |  |
| --- | --- | --- | --- |
| Opcode ( 5 bits ) | Source operand 1  (3 bits ) | Source operand 2  ( 5 bits ) | Destination operand  ( 3 bits ) |

where source operand 1 = register

source operand 2 = immediate value

destination operand = register

For Ex :

ADD R2, #3 , R5

01001 010 00011 101