**Processor Design Project**

**Interim Report**

**EE480**

Team Members: Alex Hendren & Zachary Sean McFeely

Date

# Introduction

# Instruction Set Architecture Development

## Implementation

### Add

|  |  |
| --- | --- |
| Instruction | Add |
| OPCode | 0000 |
| Flags | 00: Direct 01: Indirect 10: Immediate |
| Format | ADD FLAG OPERAND |
| Description | Adds the specified operand to the current contents of the ACC register. The Output from the ALU will be returned to the ACC register. |
| Operation | ACC ← ACC + OPERAND |
| Memory Addressing | Immediate, Direct, and Indirect |

### Sub

|  |  |
| --- | --- |
| Instruction | SUB |
| OPCode | 0001 |
| Flags | 00: Direct 01: Indirect 10: Immediate |
| Format | SUB FLAG OPERAND |
| Description | Subtracts the specified operand from the current contents of the ACC register. The Output from the ALU will be returned to the ACC register. |
| Operation | ACC ← ACC - OPERAND |
| Memory Addressing | Immediate, Direct, and Indirect |

### Logical OR

|  |  |
| --- | --- |
| Instruction | OR |
| OPCode | 0011 |
| Flags | 00: Direct 01: Indirect 10: Immediate |
| Format | OR FLAG OPERAND |
| Description | Perform a LOGICAL OR operation of contents in Acccumulator and OPERAND. The result of the operation will be stored in the accumulator. |
| Operation | ACC ← ACC | OPERAND |
| Memory Addressing | Immediate, Direct, and Indirect |

### Logical AND

|  |  |
| --- | --- |
| Instruction | AND |
| OPCode | 0100 |
| Flags | 00: Direct 01: Indirect 10: Immediate |
| Format | AND FLAG OPERAND |
| Description | Perform a LOGICAL AND operation of contents in Acccumulator and OPERAND. The result of the operation will be stored in the accumulator. |
| Operation | ACC ← ACC & OPERAND |
| Memory Addressing | Immediate, Direct, and Indirect |

### Logical Complement (COMP)

|  |  |
| --- | --- |
| Instruction | COMP |
| OPCode | 0100 |
| Flags | NULL |
| Format | COMP FLAG OPERAND |
| Description | Perform LOGICAL complement/Negation on the contents of the accumulator register. The complement will be stored in the accumulator register. |
| Operation | ACC ← NOT ACC |
| Memory Addressing | N/A |

### Multiply and Divide

|  |  |
| --- | --- |
| Instruction | MUL |
| OPCode | 0010 |
| Flags | 00: Direct 01: Indirect |
| Format | MUL FLAG OPERAND [Lower 4 bits] |
| Description | Performs multiplication on lower 4-bits of Accumulator and lower 4-bits of the operand. |
| Operation | ACC ← ACC \* OPERAND |
| Memory Addressing | Direct and Indirect |

|  |  |
| --- | --- |
| Instruction | DIV |
| OPCode | 0010 |
| Flags | 10: Direct 11: Indirect |
| Format | DIV FLAG OPERAND [Lower 4 bits] |
| Description | Performs division on lower 4-bits of Accumulator and lower 4-bits of the operand. |
| Operation | ACC ← ACC / OPERAND |
| Memory Addressing | Direct and Indirect |

### Arithmetic Left/Right Shift

|  |  |
| --- | --- |
| Instruction | SHFT |
| OPCode | 0101 |
| Flags | 00: Left Shift in 0 01: Left Shift in 1 |
| Format | SHFT FLAG OPPERAND |
| Description | Performs a LEFT SHIFT operation on the contents of the accumulator register. The value of the bit shifted in is dependent upon the provided flag. Any number of bits less than 7. |
| Operation | ACC ← ACC [N-2:0 + (FLAG BIT)\*OPERAND] |
| Memory Addressing | N/A |

|  |  |
| --- | --- |
| Instruction | SHFT |
| OPCode | 0101 |
| Flags | 10: Right Shift in 0 11: Right Shift in 1 |
| Format | SHFT FLAG OPERAND |
| Description | Performs a RIGHT SHIFT operation on the contents of the accumulator register. The value of the bit shifted in is dependent upon the provided flag. Shift by any number of bits less than 7. |
| Operation | ACC ← ACC [OPERAND\*(FLAG BIT) + N-1:1] |
| Memory Addressing | N/A |

### Conditional Branch

|  |  |
| --- | --- |
| Instruction | BRA |
| OPCode | 0110 |
| Flags | 00: Branch if equal |
| Format | BRA FLAG OPERAND |
| Description | Performs a comparison between the Accumulator and XXX. If true, jump to instruction pointed to by OPERAND. |
| Operation | IP ←IP + OPERAND |
| Memory Addressing | N/A |

### Unconditional Jump

|  |  |
| --- | --- |
| Instruction | JMP |
| OPCode | 0111 |
| Flags | NULL |
| Format | JUMP OPERAND |
| Description | Executes an unconditional branch. When encountered the instruction pointer is adjusted to the operand target memory address. |
| Operation | IP ←IP + OPERAND |
| Memory Addressing | N/A |

### Jump/Return to/from a Subroutine

|  |  |
| --- | --- |
| Instruction | RTS |
| OPCode | 1000 |
| Flags | NULL |
| Format |  |
| Description | Transfer program control to the address located in the implied return address. Return is made to the instuction that follows the CALL instruction |
| Operation | IP ← Implied Address |
| Memory Addressing | NULL |

### Return from Interrupt Service Routine

|  |  |
| --- | --- |
| Instruction | RTI |
| OPCode | 1001 |
| Flags | NULL |
| Format | RTI |
| Description | Return from an Interrupt Subroutine to the position the PC was at before the interrupt |
| Operation | IP ← Return Address |
| Memory Addressing | NULL |

### LOAD Accumulator from RAM

|  |  |
| --- | --- |
| Instruction | LOAD |
| OPCode | 1010 |
| Flags | 00: Direct 01: Immediate |
| Format | LOAD FLAG OPERAND |
| Description | **Direct Mode** Copy the contents of the address in RAM specified by the source operand to the accumulator.  **Immediate Mode:** Copy the 8-bit immediate source operand to the accumulator. |
| Operation | **Direct Mode** ACC ← memory (OPERAND)  **Immediate Mode:** ACC ← Immed OPERAND |
| Memory Addressing | Direct, Immediate |

### STORE Accumulator to RAM

|  |  |
| --- | --- |
| Instruction | STORE |
| OPCode | 1011 |
| Flags | 00: Direct 01: Indirect |
| Format | STORE FLAG OPERAND |
| Description | **Direct Mode** Copy the accumulator to the destination memory address specified by the operand **Indirect Mode:** Copy the accumulator to the desitnation memory address found at the memory address of the operand |
| Operation | **Direct Mode** memeory(OPERAND) ← ACC **Indirect Mode:** Memory{ memory( OPERAND) } ← ACC |
| Memory Addressing | Direct, Indirect |

### INPUT Data Word to RAM

|  |  |
| --- | --- |
| Instruction | INPUT |
| OPCode | 1100 |
| Flags |  |
| Format |  |
| Description | Input a data word to RAM |
| Operation |  |
| Memory Addressing |  |

### OUTPUT Data Word from RAM

|  |  |
| --- | --- |
| Instruction | OUTPUT |
| OPCode | 1101 |
| Flags |  |
| Format |  |
| Description | Output data word from RAM |
| Operation |  |
| Memory Addressing |  |

### LOAD Mask Register of HVPI

|  |  |
| --- | --- |
| Instruction | LOAD Mask Register of HVPI |
| OPCode | 1110 |
| Flags |  |
| Format |  |
| Description |  |
| Operation |  |
| Memory Addressing |  |

### NOP

|  |  |
| --- | --- |
| Instruction | NOP |
| OPCode | 1111 |
| Flags | NULL |
| Format | NOP |
| Description | Performs no operation. |
| Operation | NULL |
| Memory Addressing | NULL |

# Accumulator Architecture Development

## Architecture Overview

## Instruction Implementation

### Add

### Sub

### Logical OR

### Logical AND

### Logical Complement (COMP)

### Multiply and Divide

### Arithmetic Left/Right Shift

### Conditional Branch

### Unconditional Jump

### Jump/Return to/from a Subroutine

### LOAD Accumulator from RAM

### STORE Accumulator to RAM

### INPUT Data Word to RAM

### OUTPUT Data Word from RAM

### LOAD Mask Register of HVPI

### NOP

# Conclusion

# References