Computer Organization

- ALU
- Registers
  - Accumulator (AR)
  - Program Counter (PC)
  - Instruction Register (IR)
  - General Purpose Registers
- Memory
- Operation Decoder

Computer Organization

- Clock
  - Timing Counter (TC)
  - Counts the Number of Clock Cycles
- Hardware/Firmware Implemented Instructions
- Instruction Format
  - OP Code Field - Specifies the Instruction to be Executed
  - Address Field - Specifies Memory Location or Register
  - Mode Field - Specifies the Way in Which the Address Is To Be Interpreted

Computer Operation

- **Fetch**
  - **AR ← PC**
  - **DR ← M[AR]**
  - **PC ← PC + 1**
- **Execute**
  - **IR ← OpCode: Determine Next Microinstruction in the Sequence**
  - **Execute Microinstruction**

OP Codes

- **Often Referred to By a *Mnemonic***
- **Specifies a Useful Operation That May Encompass Several Clock Cycles**
- **Opcodes Are the Binary Equivalent of the Mnemonic**
- **The Opcodes Must Have Enough Bits to Represent All Desired Operations ($\log_2 n$)**

Data Transfer Instructions

- Instructions to Move Data Between Registers and Memory
- Examples
  - Load (LD)
  - Store (ST)
  - Move (MOVE)

Data Manipulation Instructions

- Instructions That Alter Data
- Examples
  - Add (ADD)
  - Increment (INC)
  - And (AND)
  - Or (OR)
  - Shift right (SHR)
  - Rotate Left (ROL)

Control Instructions

- Instructions That Alter the Flow of a Program
- Examples
  - Jump (JMP)
  - Subroutine (SUB)
  - Branch if Zero (BZ)
  - Branch if Positive (BP)

Types of Instructions

- One Address
  - Often Uses an Implied Accumulator Register
  - Example: LDA, ADD X
- Two Address (MOVE A,R1)
- Three Address (ADD A, B, R1)

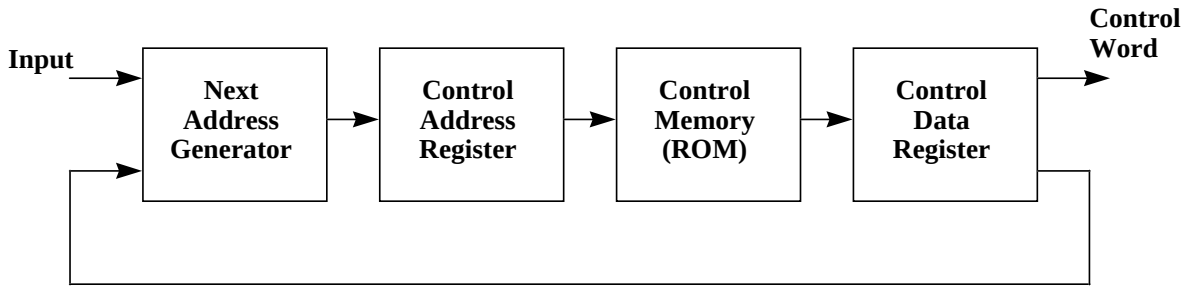
Addressing Modes

- **Direct**
  - **the address is contained in the address field**
  - **the size of the memory is limited by the size of the address field**
- **Indirect**
  - **Content of the Memory Location Contained in the Address Field Points to The Actual Address**
  - **Allows for a Larger Memory Because the Address Fields Can Be Larger**
  - **Allows for Efficient Address Manipulation**
- **Indexed**
  - **Content of the Memory Field is Added to the Contents of the *Index Register***
  - **Allows for Flexible Relative Addressing**

Control System Design

- **A Programmer Needs a Logical Structure and Instructions**
- **A Hardware Designer has Microoperations**
- **The Bridge Between These is a Microprogram**

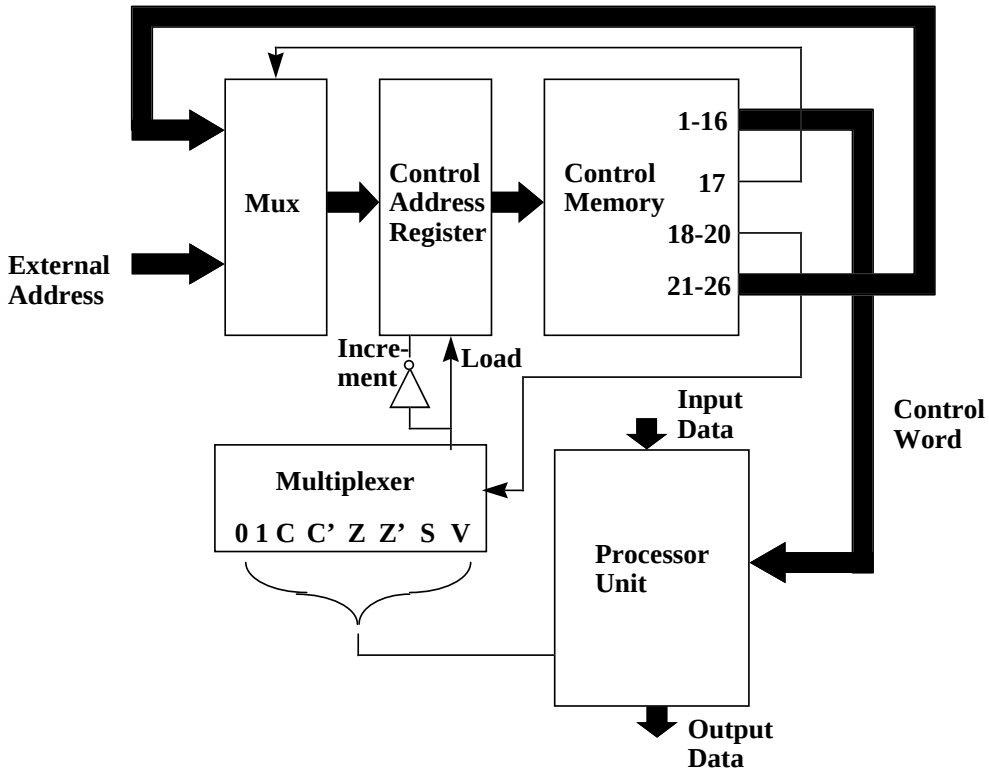
Structure of a Microprogrammed Control Unit



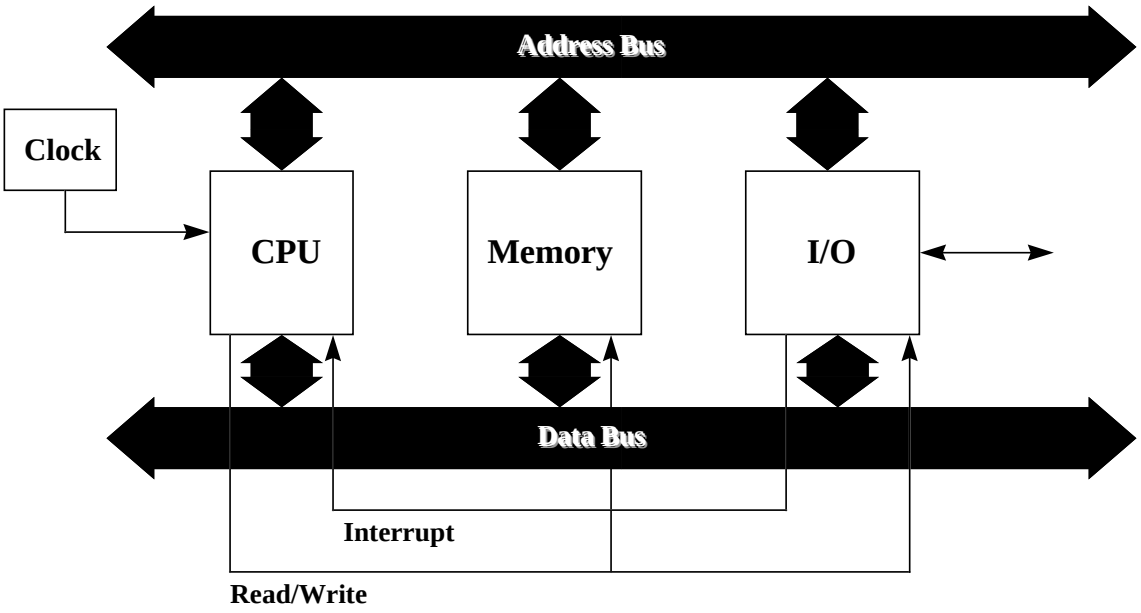
Example

- General
  - 16 Bits for Processor Control (Bits 1-16; A,B,D,F,H As Before)
  - One Bit for Address Source Selection (bit 17)
  - Three Bits For Status Bit Select (Bits 18-20)
  - Six Bits for the Next Address (Bits 21-26)
- 2^6=64 Microinstructions Are Possible
- CAR Can be Loaded or Incremented, Depending on the Condition

Example



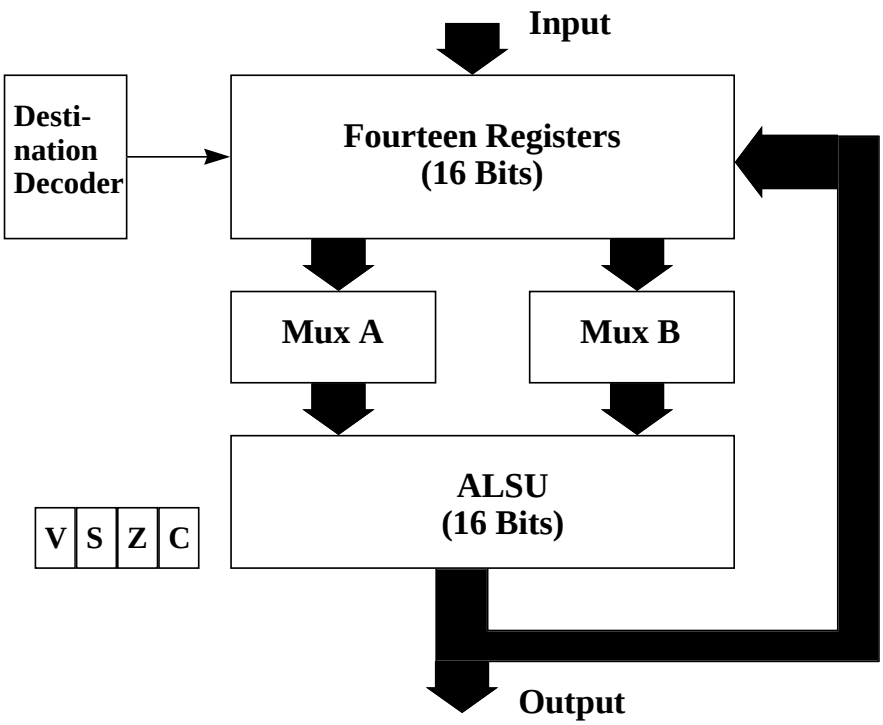
A Simple Computer Design



A Simple Computer Design

- I/O Interface
- 16 Bit Data Bus
- 16 Bit Address Bus

CPU Components



CPU Components

- ALSU
- Registers
- Internal Busses
- Status Bits

ALSU

- Two 16 Bit Input Busses
- Four Units Multiplexed Together
- Each Unit Always Operates on Both Operands
- The Operation is Determined by the Selection Lines S_0 and S_1
- Operation is Selected by a 4:1 Multiplexer (S_2 and S_3)
- 16 Operations are Possible on the Output
- Example 0110 => F=A B
- Requires a Five Bit Control Word

ALSU Function Table

Operation Select				Function	
S_3	S_2	S_1	S_0	$C_{in} = 0$	$C_{in} = 1$
0	0	0	0	F=A	F=A+1
0	0	0	1	F=A+B	F=A+B+1
0	0	1	0	F=A-B -1	F=A-B
0	0	1	1	F=A-1	F=A
0	1	0	0	F=AB	
0	1	0	1	F=A+B	
0	1	1	0	F=A XOR B	
0	1	1	1	F=A’	
1	0	0	0	F=shr A	
1	0	0	1	F=ror A	
1	0	1	0	F=ror A w. Carry	
1	0	1	1	F=asr A	
1	1	0	0	F=shl A	
1	1	0	1	F=rol A	
1	1	1	0	F=rol A w. Carry	
1	1	1	1	F=asl A	

Registers

- Fourteen Total Registers
- PC
- Six General Purpose Registers
- Seven Special Purpose Registers
  - Index Register
  - Stack Pointer
  - Source Register
  - Destination Register
  - Temporary Register
  - Two Constant Registers (Zero and N=16)

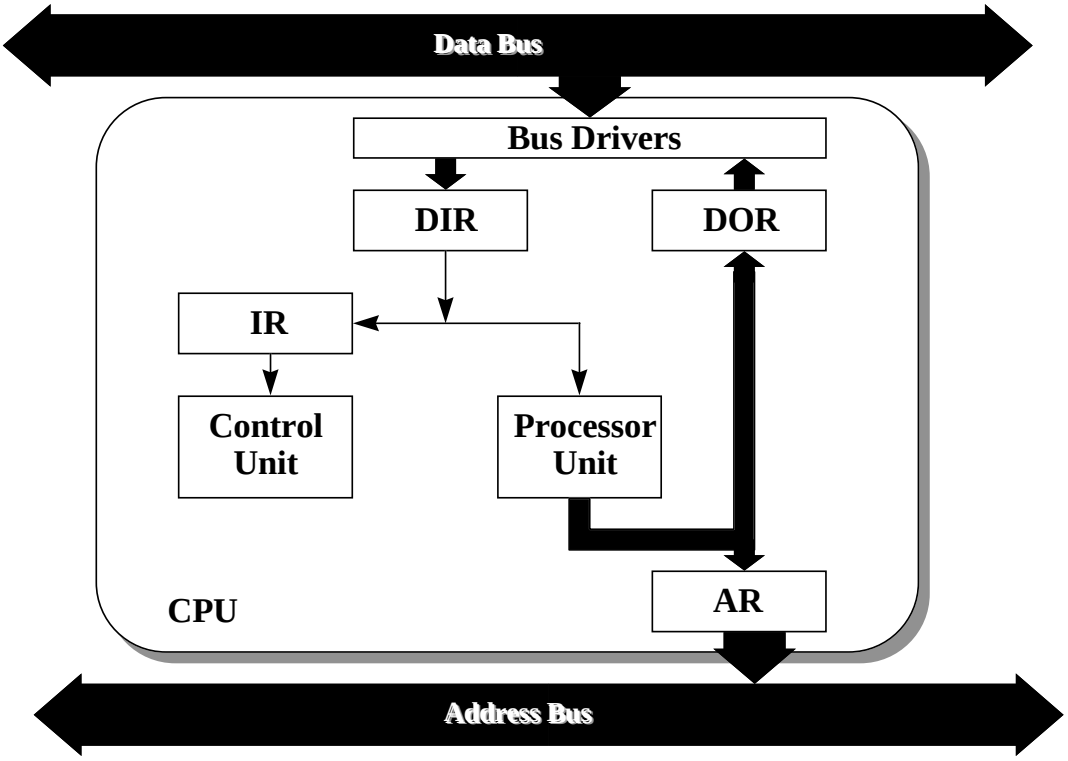
Three Internal Busses

- One for Each Operand
  - Attached to One of the Registers via a Multiplexer
  - One Mux for Each Operand Bus
- One For the Result
  - Destination is One of the Registers
  - May Also Be External

Control Word

- 17 Bits Long
- Five Bits for ALSU (Four Control + C_{in})
- Four Bits for Mux A
- Four Bits for Mux B
- Four Bits for Destination Decoder

Organization of the CPU



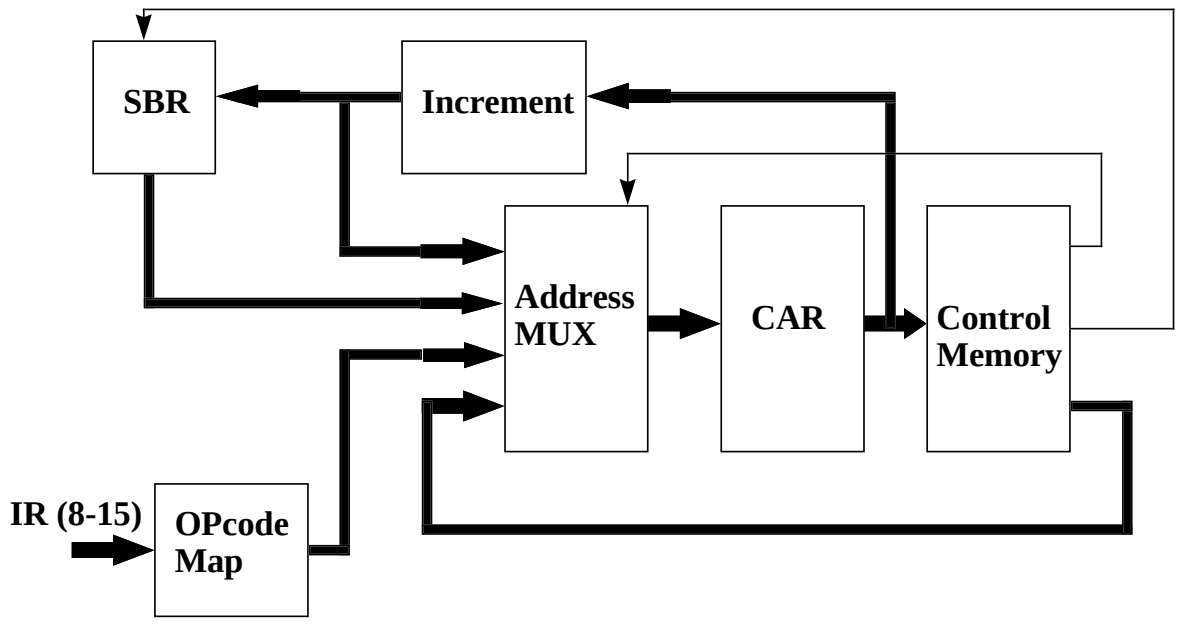
CPU

- Components
  - Processor Unit
  - Memory
  - Control Unit
  - Buffers/Registers
  - Busses
- Busses
  - External to the Processor
  - Data Bus (16 bits)
    - Direction Must be Mediated
    - Read/Write Line (From/To Memory))
  - Address Bus (16 bits)

CPU Buffers and Registers

- Data Input Register
  - Input to IR or Processor
  - Data from Memory or From the Outside World
- Data Output Register - Output to Memory or Oustide World
- Address Register - Current Memory Address
- Instruction Register - OPcode of the Current Instruction

Control Unit Organization



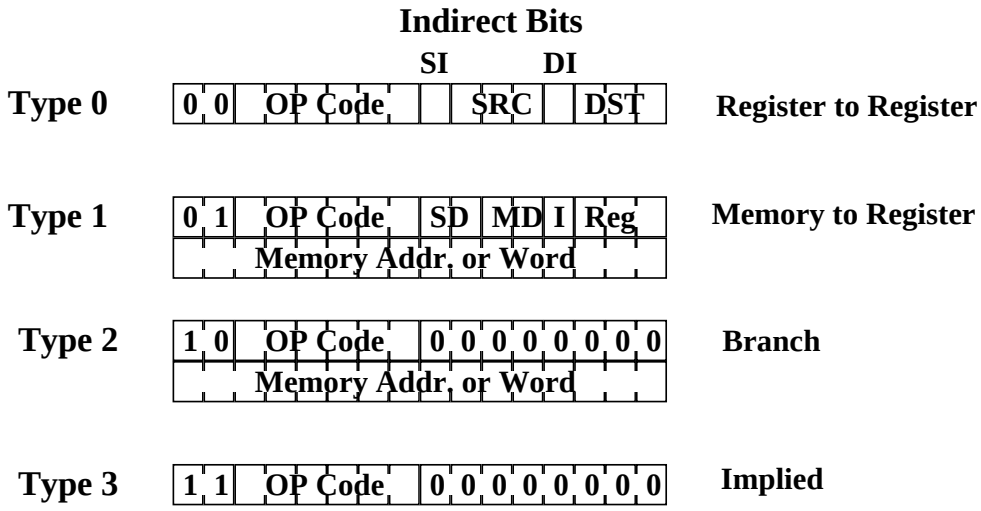
Control Unit

- Components
  - Subroutine Register
  - OPCode Map
  - Incrementer
  - Address Mux
  - Control Address Register
  - Control Memory
- Subroutine Register
  - Used in Subroutine Call
  - Stores Return Address

Control Unit

- Opcode Map
  - Maps OPcode Into Control Memory Address
  - May be a ROM
- Incrementer - Increments Address
- Address Mux
  - Selects Source of Address
  - Controlled by Control Memory (CS field)
- Control Address Register - Holds the Memory Address
- Control Memory - Contains the Microcode

Instruction Types



Type 0

- Register-Register
- One Word Instruction
- Format
  - Bits 0-2: Destination Register
  - Bit 3: Indirect Bit (0=Direct, 1=Indirect)
  - Bits 4-6: Source Register
  - Bit 7: Indirect Bit
  - Bits 8-13: OPCode
  - Bits 14-15: 00 (Indicates Register-Register Operation)

Type 1

- Memory-Register
- Two Word Instruction (Except for One Operand Instructions with Operand in Register)
- Format
  - Bits 0-2: Register
  - Bit 3: Indirect Bit
  - Bits 4-5: Addressing Mode
    - Immediate: W is the Operand
    - Direct
    - Indirect
    - Index
  - Bit 6-7: Source/Destination and Number of Operands
    - Memory or Register
    - One or Two Operands

Type 1

- Format
  - Bits 8-13: OPCode
  - Bits 14-15: 01 (Indicates Memory-Register Operation)
  - Second Word for Memory Address or Intermediate Operand

Type 2

- Branch
- Two Word Instruction
- Format
  - Bits 0-7: 0
  - Bits 8-13: OPCode
  - Bits 14-15: 10 (Indicates Branch Operation)
- Second Word Contains the Branch Address

Type 3

- Implied Mode
- Operand Either Does Not Exist or is Implicit in the Instruction
- Example: NOP
- One Word
- Format
  - Bits 0-7: 0
  - Bits 8-13: OP code
  - Bits 14-15: 11

An Assembly Language Instruction: ADD

- Adds Two Numbers
- May be Type 0 or Type 1
  - Type 0 if the Operands are in Registers
  - Type 1 if One of Them is in Memory
- Example of Type 0: ADD R5,R2
  - Operation: R2← R2+R5
  - Machine Code (0952H)
    - Bits 0-2: 010 (Register 2)
    - Bit 3: Indirect Bit = 0 (Direct)
    - Bits 4-6: 101 (Register 5)
    - Bit 7: Indirect Bit = 0 (Direct)
    - Bits 8-13: OPCode = 001001
    - Bits 14-15: 00 (Indicates Register-Register Operation)
  - Note That All Type 1 ADD Instructions Have 09H As First Byte

Alternate Form of ADD

- If We Used Indirect in R2: ADD R5,(R2)
  - Operation: M[R2] M[R2]+R5
  - Machine Code:
    - Bits 0-2: 010 (Register 2)
    - Bit 3: Indirect Bit = 1 (Direct)
    - Bits 4-6: 101 (Register 5)
    - Bit 7: Indirect Bit = 0 (Direct)
    - Bits 8-13: OPCode = 001001
    - Bits 14-15: 00 (Indicates Register-Register Operation)
  - Alternatively: 095AH

Example of a Type 1 Addition: ADD TEMP,R2

- R2←R2+M[TEMP]
- Machine Code
  - Bits 0-2: 010
  - Bit 3: Indirect bit = 0
  - Bits 4-5: Addressing Mode = 00
  - Bit 6-7: Source/Destination and Number of Operands = 00
  - Bits 8-13: OPCode = 001001
  - Bits 14-15: 01
- Machine Code: 4902H

Notes

- Many Other Instructions Exist as Well
- We Will Examine the Intel 8080 Later
- The Software That Translates Assembly Code to Machine is an Assembler

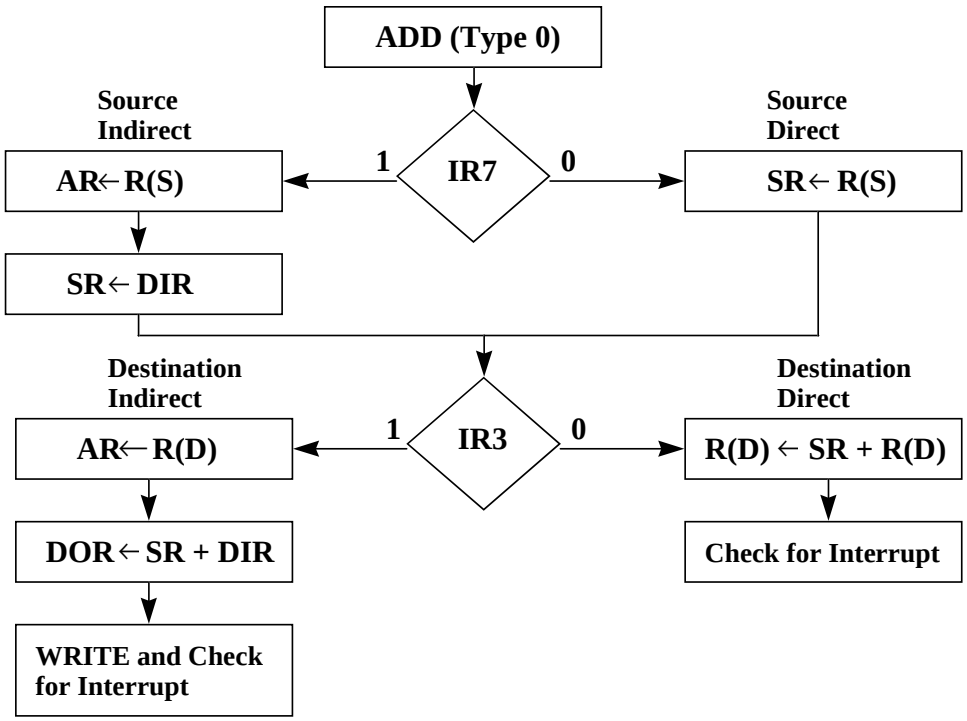
Microinstructions

- 23 bits
- 17 for ALSU
  - Operand A Select (4)
  - Operand B Select (4)
  - Destination Select (4)
  - ALSU Function Select (5)
- Six for Other Functions
  - Two for Control Sequence
    - Distinguishes Microinstruction Formats
    - Controls the Address Mux in the Control Unit
  - Four for Miscellaneous Microoperations

Example: R2←R2+R5

- Refer to Mano, Figure 10-8(b) (p. 349)
- Control Sequence (CS) = 00
- Register Select “A” (AS) = 0010
- Register Select “B” (BS) = 0101
- Destination Select (DS) = 0010
- Function Control (FC) = 00010
- Miscellaneous (MS) = 0000
- Hex code: 04A620
- Microprograms Map OPcode Semantics into Microoperation Sequences

Microprogram Flowchart for Type 0 ADD



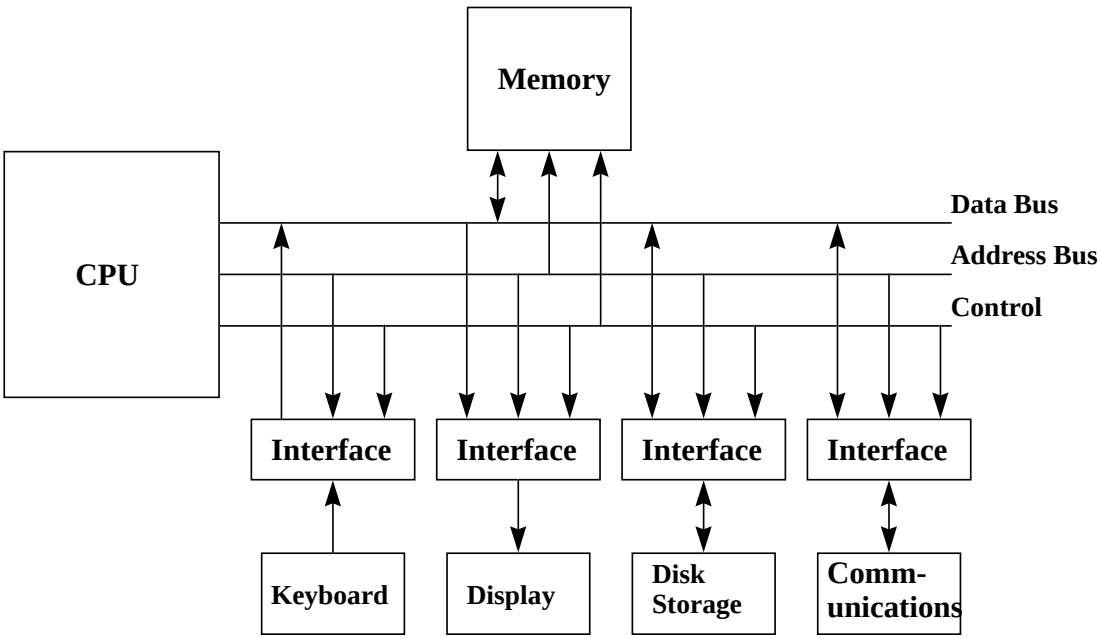
Program Status Word (PSW)

- General
  - Contains Important CPU Status Indications
  - Need Sufficient Information to Restore Processor Context (Status)
- Typical Contents
  - Program Counter
  - Stack Pointer
  - Accumulator Value
  - Index Register(s)
  - Processor Status Register
  - Instruction Register

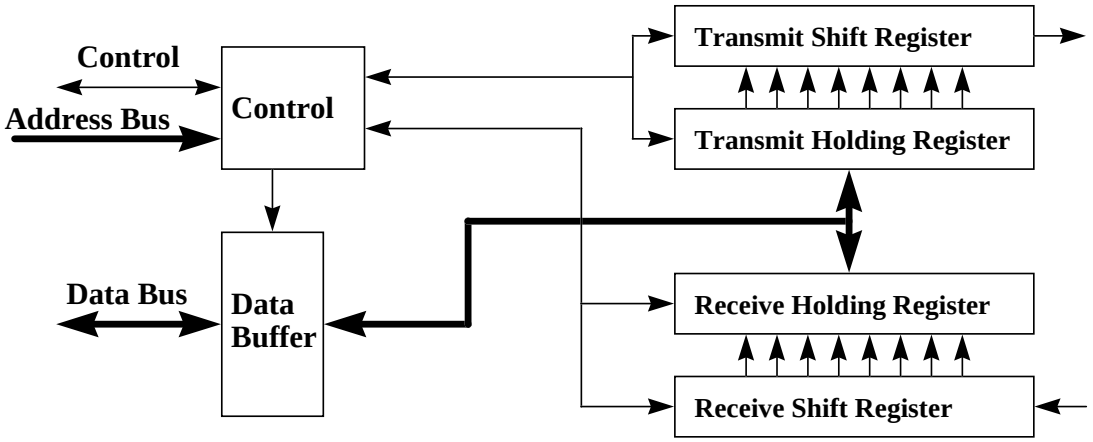
Stack

- Special Type of Memory
- Operations
  - PUSH
  - POP

Input/Output for Communications



Serial Communications Devices



Computer Architecture - Intel 8080

- Defines Only the CPU
- Physical Features
  - 40 Pin DIP Package (See Handout)
  - 8 Bit Data Bus
  - 16 Bit Address Bus
  - Control Pins

Logical Features

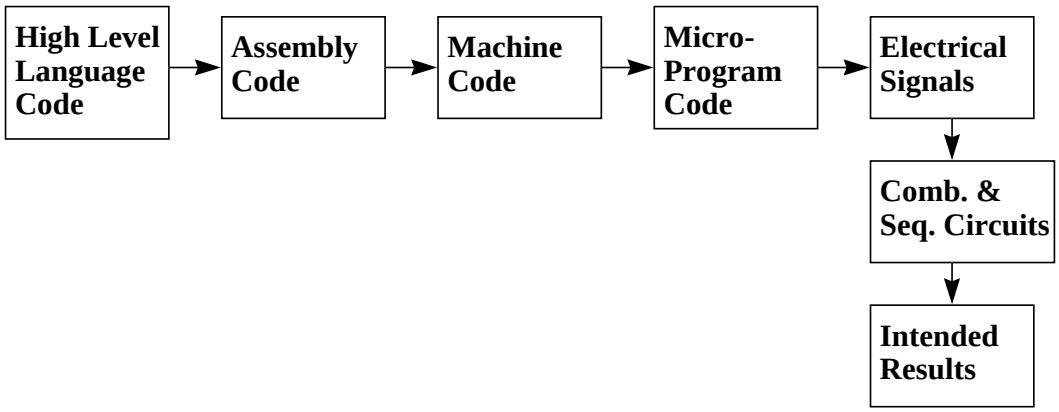
- 8 Bit ALU
- Registers
  - ACC
  - PC
  - Stack Pointer
  - IR
  - 6 Working Registers
    - Used in Pairs
    - B-C
    - D-E
    - H-L
  - 2 Temporary Registers
- Organization (See Handout)

Instruction Format

- 8 Bit Opcode
- 256 Instructions
- 8080 Instructions (See Handout)
- 8080 Microinstructions (See Handout)

Summary of Digital and Computer Section

- **Digital Computers Consist of Sequential and Combinational Circuits**
- **General Purpose Devices That Can Be Programmed**
- **Control is Often Implemented Via Microprograms**



Modem Revisited