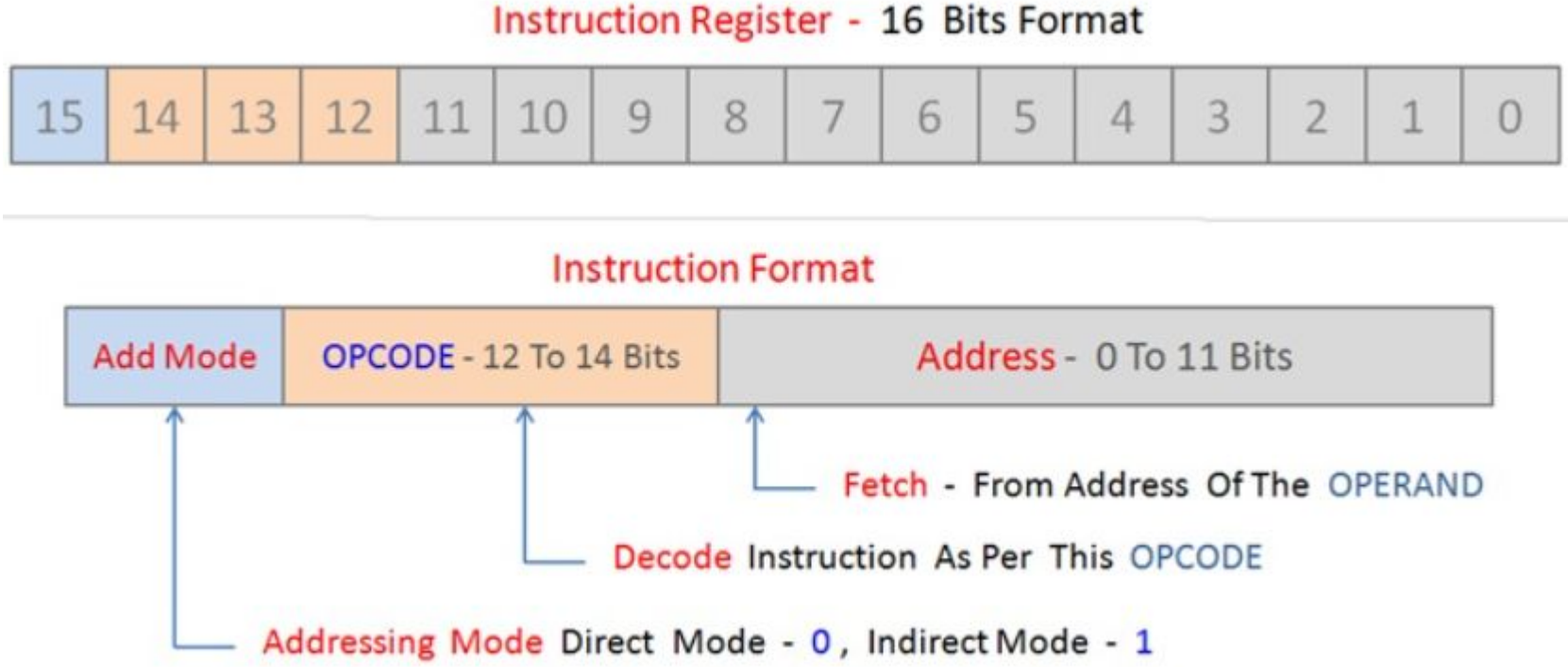

Instruction Register - 16 Bits Format
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Instruction Format
Add Mode
OPCODE - 12 To 14 Bits
Address - 0 To 11 Bits
Fetch - From Address Of The OPERAND
Decode Instruction As Per This OPCODE
Addressing Mode Direct Mode - 0, Indirect Mode - 1

Types of addressing formats

Zero-Address Instructions

$$X = (A + B) * (C + D)$$

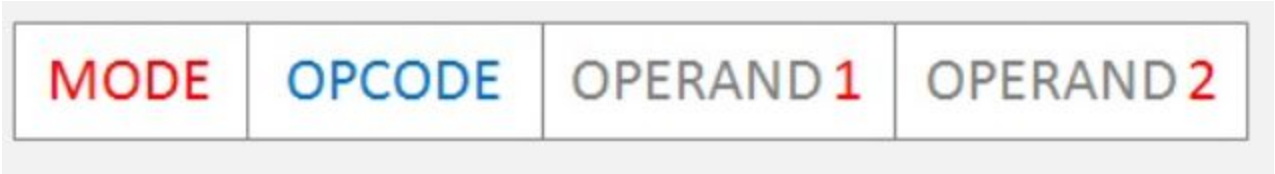
ASSEMBLY LANGUAGE	REGISTER OPERATION
PUSH A	[TOS ← A]
PUSH B	[TOS ← B]
ADD	[TOS ← (A + B)]
PUSH C	[TOS ← C]
PUSH D	[TOS ← D]
ADD	[TOS ← (C + D)]
MUL	[TOS ← (C + D) * (A + B)]
POP X	[M [X] ← TOS]

One-Address Instructions : accumulator (AC) register for data manipulation.

$X = (A + B) * (C + D)$

ASSEMBLY LANGUAGE INSTRUCTION	REGISTER OPERATION INSTRUCTION
LOAD A	[AC ← M [A]]
ADD B	[AC ← A [C] + M [B]]
STORE T	[M [T] ← AC]
LOAD C	[AC ← M [C]]
ADD D	[AC ← AC + M [D]]
MUL T	[AC ← AC * M [T]]
STORE X	[M [X] ← AC]

Two-Address Instructions



X = (A + B) ∗ (C + D)

ASSEMBLY LANGUAGE INSTRUCTION	REGISTER OPERATION INSTRUCTION
MOV R1, A	[R1 ← M [A]]
ADD R1, B	[R1 ← R1 + M [B]]
MOV R2, C	[R2 ← M [C]]
ADD R2, D	[R2 ← R2 + M [D]]
MUL R1, R2	[R1 ← R1∗R2]
MOV X, R1	[M [X] ← R1]

Three-Address Instructions

$$X = (A + B) * (C + D)$$

ASSEMBLY LANGUAGE	REGISTER OPERATION
ADD R1, A, B	[R1 ←M [A] + M [B]]
ADD R2, C, D	[R2 ←M [C] + M [D]]
MUL X, R1, R2	[M [X] ← R1 ∗R2]

Instruction Codes

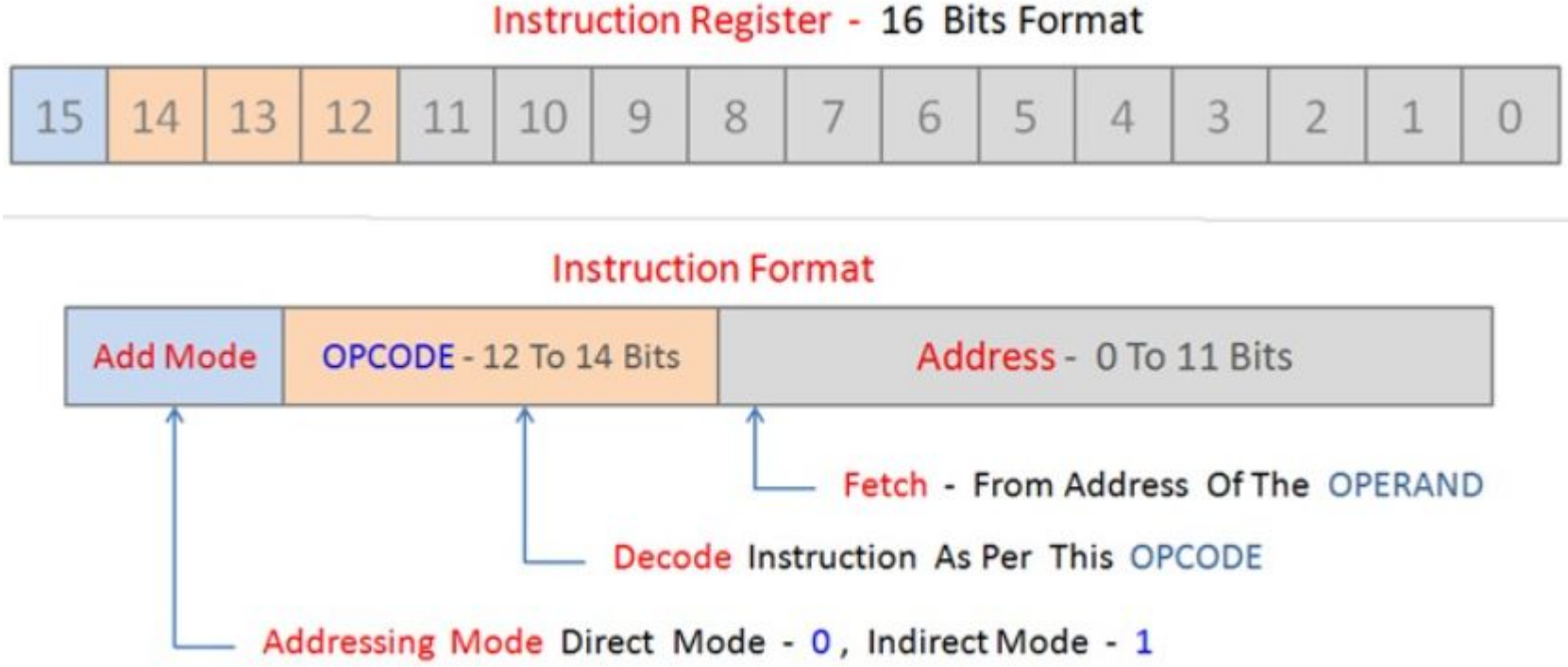
- A set of instructions that specify the operations, operands, and the sequence by which processing has to occur.
- An instruction code is a group of bits that tells the computer to perform a specific operation part.

Addressing Modes

1. Implied Mode
2. Immediate Mode
3. Register Mode
4. Register Indirect Mode
5. Auto increment or Auto decrement Mode
6. Direct Address Mode
7. Indirect Address Mode
8. Relative Address Mode
9. Indexed Addressing Mode
10. Base Register Addressing Mode

Con

Register symbol	Number of bits	Register name	Register Function----------
DR	16	Data register	Holds memory operands
AR	12	Address register	Holds address for memory
AC	16	Accumulator	Processor register
IR	16	Instruction register	Holds instruction code
PC	12	Program counter	Holds address of instruction
TR	16	Temporary register	Holds temporary data
INPR	8	Input register	Holds input character
OUTR	8	Output register	Holds output character


Instruction Register - 16 Bits Format
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Instruction Format
Add Mode
OPCODE - 12 To 14 Bits
Address - 0 To 11 Bits
Fetch - From Address Of The OPERAND
Decode Instruction As Per This OPCODE
Addressing Mode Direct Mode - 0, Indirect Mode - 1

Types of addressing formats

Zero-Address Instructions

$$X = (A + B) * (C + D)$$

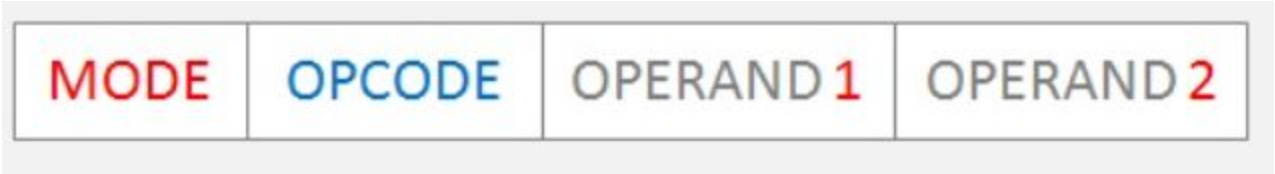
ASSEMBLY LANGUAGE	REGISTER OPERATION
PUSH A	[TOS ← A]
PUSH B	[TOS ← B]
ADD	[TOS ← (A + B)]
PUSH C	[TOS ← C]
PUSH D	[TOS ← D]
ADD	[TOS ← (C + D)]
MUL	[TOS ← (C + D) * (A + B)]
POP X	[M [X] ← TOS]

One-Address Instructions : accumulator (AC) register for data manipulation.

$X = (A + B) * (C + D)$

ASSEMBLY LANGUAGE INSTRUCTION	REGISTER OPERATION INSTRUCTION
LOAD A	[AC ← M [A]]
ADD B	[AC ← A [C] + M [B]]
STORE T	[M [T] ← AC]
LOAD C	[AC ← M [C]]
ADD D	[AC ← AC + M [D]]
MUL T	[AC ← AC * M [T]]
STORE X	[M [X] ← AC]

Two-Address Instructions



X = (A + B) ∗ (C + D)

ASSEMBLY LANGUAGE INSTRUCTION	REGISTER OPERATION INSTRUCTION
MOV R1, A	[R1 ← M [A]]
ADD R1, B	[R1 ← R1 + M [B]]
MOV R2, C	[R2 ← M [C]]
ADD R2, D	[R2 ← R2 + M [D]]
MUL R1, R2	[R1 ← R1∗R2]
MOV X, R1	[M [X] ← R1]

Three-Address Instructions

$$X = (A + B) * (C + D)$$

ASSEMBLY LANGUAGE	REGISTER OPERATION
ADD R1, A, B	[R1 ←M [A] + M [B]]
ADD R2, C, D	[R2 ←M [C] + M [D]]
MUL X, R1, R2	[M [X] ← R1 ∗R2]

Instruction Codes

- A set of instructions that specify the operations, operands, and the sequence by which processing has to occur.
- An instruction code is a group of bits that tells the computer to perform a specific operation part.

Addressing Modes

1. Implied Mode
2. Immediate Mode
3. Register Mode
4. Register Indirect Mode
5. Auto increment or Auto decrement Mode
6. Direct Address Mode
7. Indirect Address Mode
8. Relative Address Mode
9. Indexed Addressing Mode
10. Base Register Addressing Mode

Con

Register symbol	Number of bits	Register name	Register Function----------
DR	16	Data register	Holds memory operands
AR	12	Address register	Holds address for memory
AC	16	Accumulator	Processor register
IR	16	Instruction register	Holds instruction code
PC	12	Program counter	Holds address of instruction
TR	16	Temporary register	Holds temporary data
INPR	8	Input register	Holds input character
OUTR	8	Output register	Holds output character

Three-Address Instructions

$$X = (A + B) * (C + D)$$

ASSEMBLY LANGUAGE	REGISTER OPERATION
ADD R1, A, B	[R1 ←M [A] + M [B]]
ADD R2, C, D	[R2 ←M [C] + M [D]]
MUL X, R1, R2	[M [X] ← R1 ∗R2]

Instruction Codes

- A set of instructions that specify the operations, operands, and the sequence by which processing has to occur.
- An instruction code is a group of bits that tells the computer to perform a specific operation part.

Addressing Modes

1. Implied Mode
2. Immediate Mode
3. Register Mode
4. Register Indirect Mode
5. Auto increment or Auto decrement Mode
6. Direct Address Mode
7. Indirect Address Mode
8. Relative Address Mode
9. Indexed Addressing Mode
10. Base Register Addressing Mode

Con

Register symbol	Number of bits	Register name	Register Function----------
DR	16	Data register	Holds memory operands
AR	12	Address register	Holds address for memory
AC	16	Accumulator	Processor register
IR	16	Instruction register	Holds instruction code
PC	12	Program counter	Holds address of instruction
TR	16	Temporary register	Holds temporary data
INPR	8	Input register	Holds input character
OUTR	8	Output register	Holds output character

Basic Computer Organization and Design

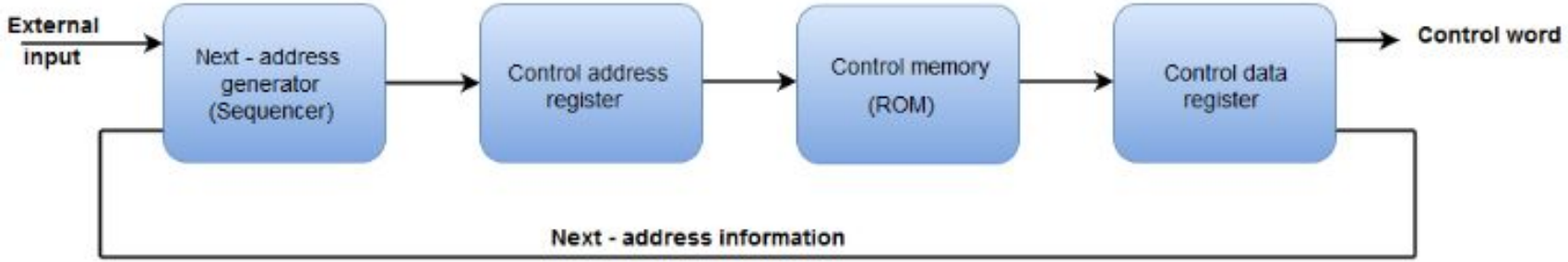
Computer Instructions

- Memory Reference
- Register Reference
- Input/Output

Micro Programmed Control

- Control Memory
- Address Sequencing

Micro-programmed Control



Computer Instructions

- Memory Reference
- Register Reference
- Input/Output

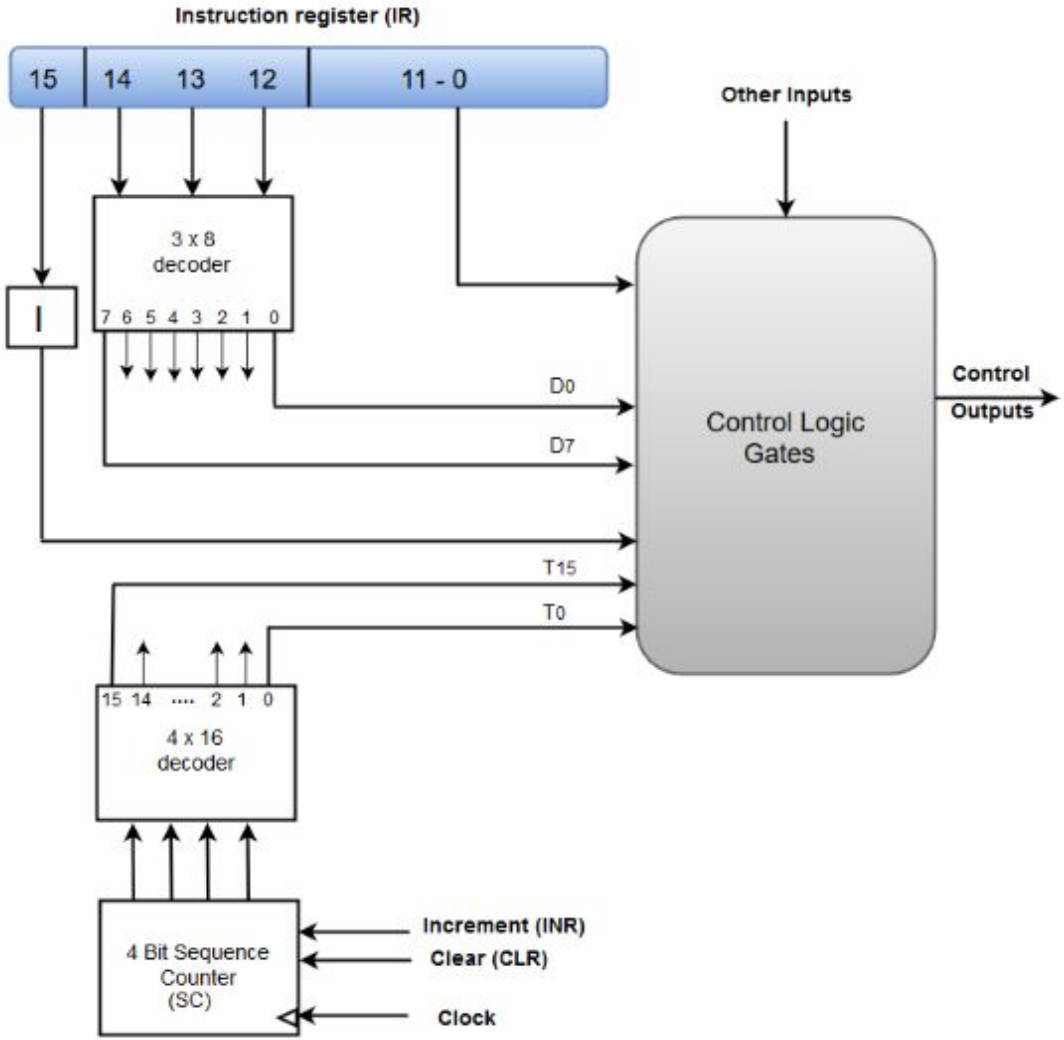
Timing and Control, Instruction cycle.

1. Fetch instruction
2. Decode
3. Fetch operand
4. Execute

Micro Programmed Control

- Control Memory
- Address Sequencing

Design of Control Unit



Micro-programmed Control