CS & IT ENGINEERING

COMPUTER ORGANIZATION AND ARCHITECTURE

Instruction & Addressing Modes

Lecture No.- 03

By- Vishvadeep Gothi sir

Recap of Previous Lecture

Topic Instruction

Topics to be Covered

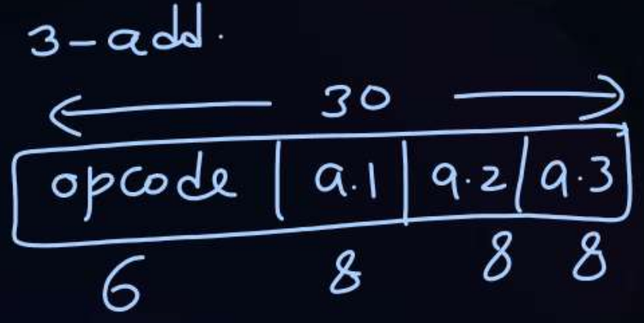
Topic Register Spill

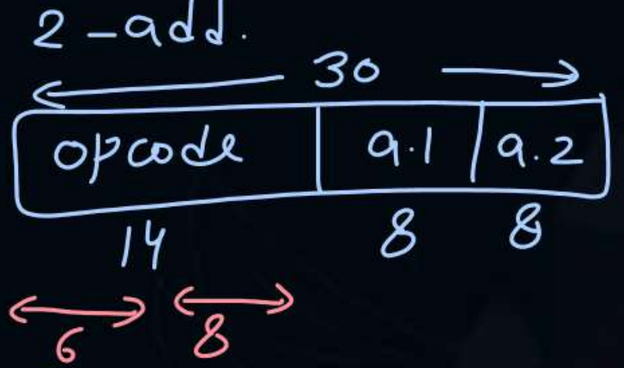
Topic Variable Length Instructions

[NAT]

Ans = $(64-x) * 2^8$

#Q. Consider a system which supports 3-address and 2-address instructions both. It has 30-bit instructions with 8-bit addresses. If there are 'x' 3-address instructions, then maximum how many 2-address instructions can be formulated?





max = $2^6 = 64$

used = x

unused = 64 − x

max = $(64-x) * 2^8$

Assume there are 512 2-add. instns in prev. Questn, then max no. of 3-add. instns are ?

Soln

$$(64 - x) * 2^8 = 512$$

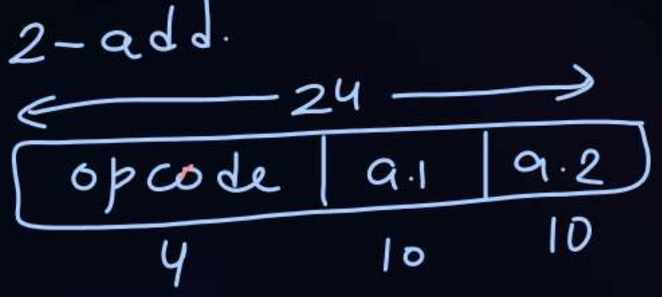
$$\boxed{x = 62}$$

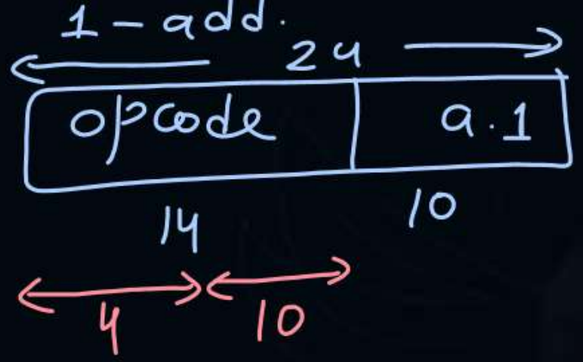
Ans = 62

[NAT]

Ans = 12

#Q. Consider a system which supports 2-address and 1-address instructions both. It has 24-bit instructions with 10-bit addresses. If there are 4096 1-address instructions then maximum how many 2-address instructions can be formulated?





$max = 2^4 = 16$

used = x

unused = (16 − x)

$(16 - x) * 2^{10} = 4096$

$16 - x = 4$

$x = 12$

[NAT]

→ Reg. ⇒ 6-bits

#Q. Consider a system with 16-bits instructions and 64 CPU registers. The System supported 2 types of instructions: Type-A and Type-B.

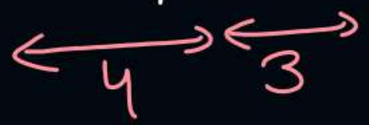
Type-A instructions have an opcode, one register operand and one immediate operand of 3-bits

Type-B instructions have an opcode, and 2 register operands.

If there are 10 Type-B instructions supported by the system then maximum how many Type-A Instructions supported by the system?

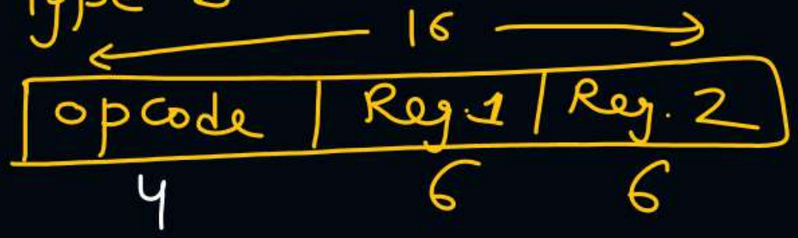
Ans = 48

Type-A



$max = 6 * 2^3 = 48$

Type B



$max\ opcodes = 2^4 = 16$

$used\ opcodes = 10$

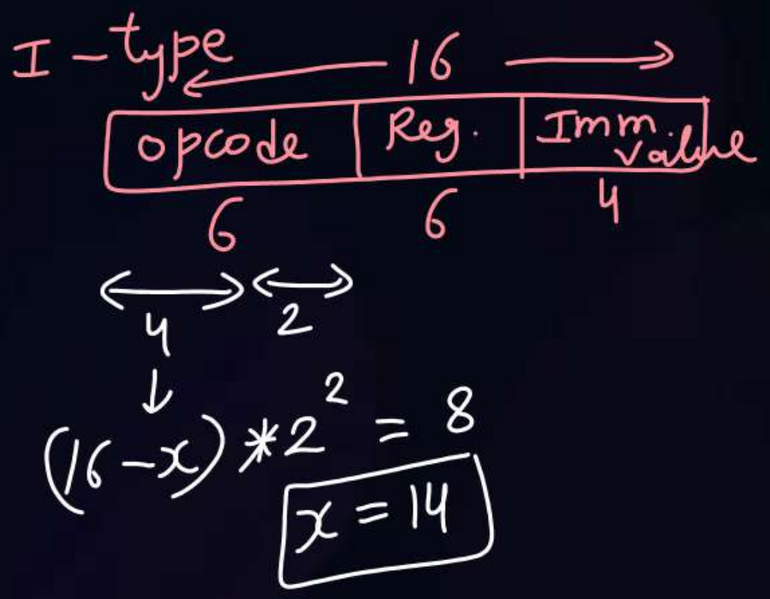
$unused\ opcodes = 6$

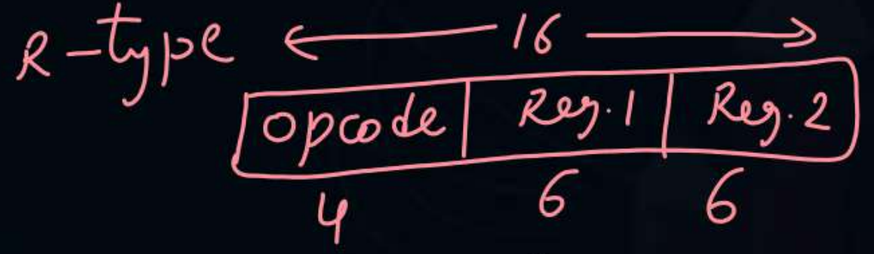
[NAT] GATE-2020

Reg. ⇒ 6-bits

Ans = 14

#Q. A processor has 64 registers and uses 16-bit instruction format. It has two types of instructions: I-type and R-type. Each I-type instruction contains an opcode, a register name, and a 4-bit immediate value. Each R-type instruction contains an opcode and two register names. If there are 8 distinct I-type opcodes, then the maximum number of distinct R-type opcodes is _____?



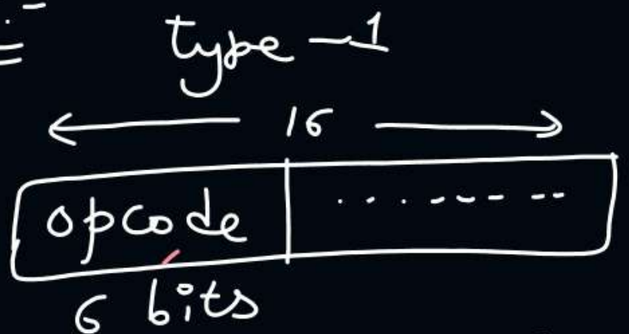


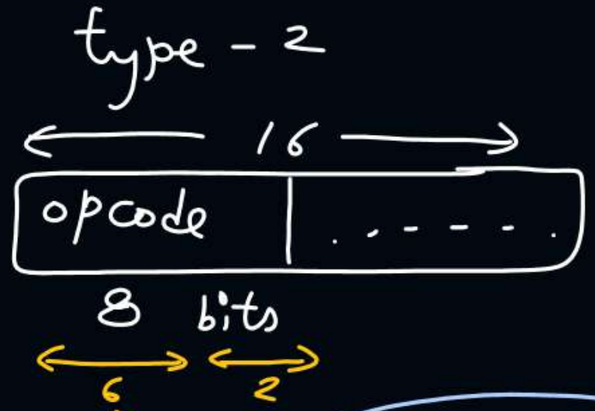
max opcodes $= 2^4 = 16$

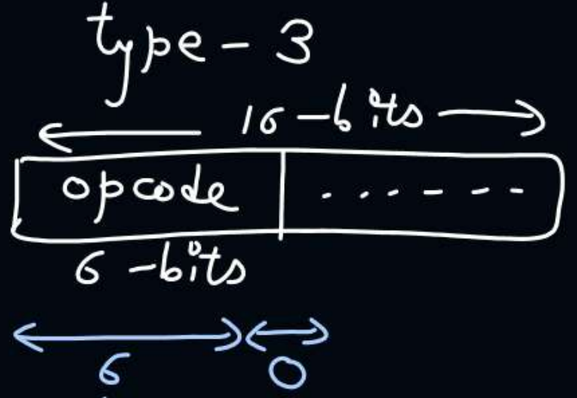
used opcodes $= x$

unused $= 16 - x$

ex:-







$max = 2^6 = 64$

used = 30

unused = 34

Given,

type 1 30 inst^ns

type 3 28 inst^ns

type 2 ⇒ —— ?

$max = 6 * 2^2 = \underline{24}$

$max = 34 * 2^0 = 34$

used = 28

unused = 6

Ans = 24

[NAT] GATE-2018

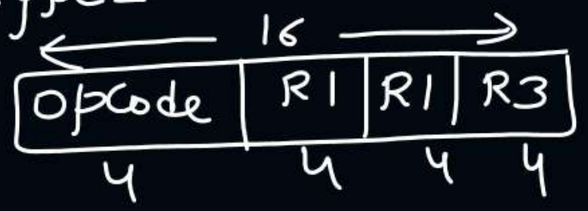
4-bits 16-bits 6-bits

#Q. A processor has 16 integer registers (R0, R1, … , R15) and 64 floating point registers (F0, F1, … , F63). It uses a 2-byte instruction format. There are four categories of instructions: Type-1, Type-2, Type-3, and Type 4. Type-1 category consists of four instructions, each with 3 integer register operands (3Rs). Type-2 category consists of eight instructions, each with 2 floating point register operands (2Fs). Type-3 category consists of fourteen instructions, each with one integer register operand and one floating point register operand (1R+1F). Type-4 category consists of N instructions, each with a floating-point register operand (1F).

The maximum value of N is _____ ?

Ans = 32

Type 1

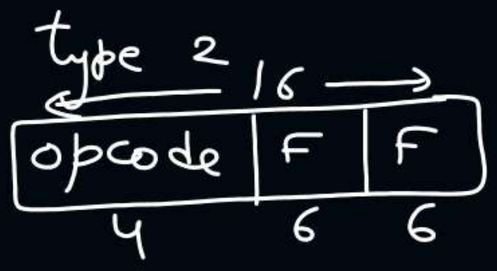


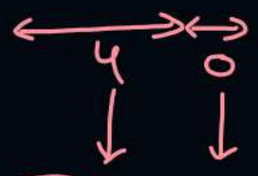
$max = 2^4 = 16$

used = 4

unused = 12

type 2



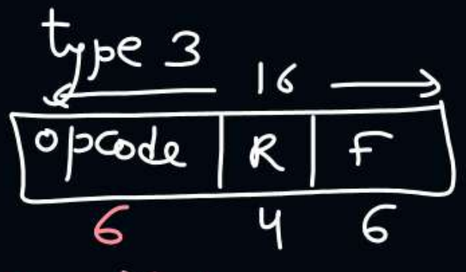


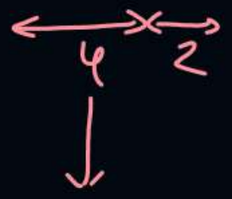
$max = 12 * 2^0 = 12$

used = 8

unused = 4

type 3



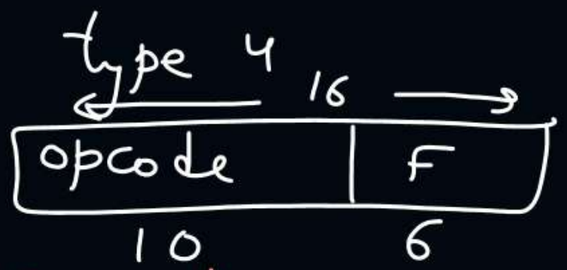


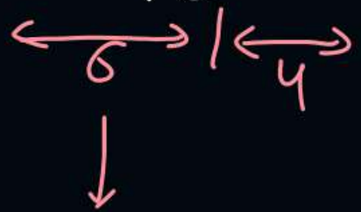
$max = 4 * 2^2 = 16$

used = 14

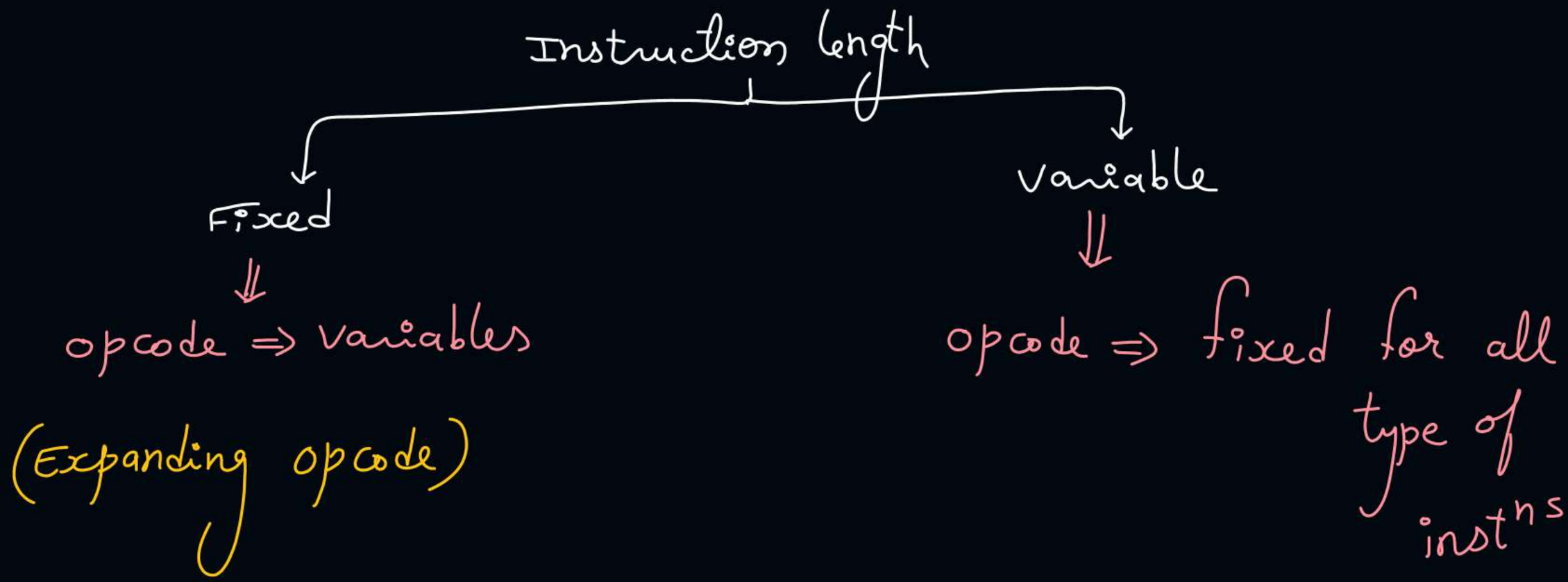
unused = 2

type 4





$max = 2 * 2^4 = 32$


Instruction length
Fixed
Variable
opcode ⇒ variables
(Expanding opcode)
opcode ⇒ fixed for all type of instns

[NAT]

Max = 40 bits
Min = 17 bits

#Q. Consider there are 3 types of instructions in system:

1. Register Operand instructions: One opcode and 2 registers
2. Memory Operand instructions: One opcode, 1 register and 1 memory address
3. Immediate Operand Instructions: One opcode, 1 register and 1 immediate operand

Number of registers = 64 → Reg. = 6-bits

Number of bits in immediate operand = 10-bits

Memory size = 512Mbytes (byte addressable) = $2^9 \cdot 2^{20}$ B = 2^{29} B ⇒ add. = 29 bits

Total Instructions:

1. Reg Operand type: 10
2. Memory Operand type : 12
3. immediate Operand type : 4

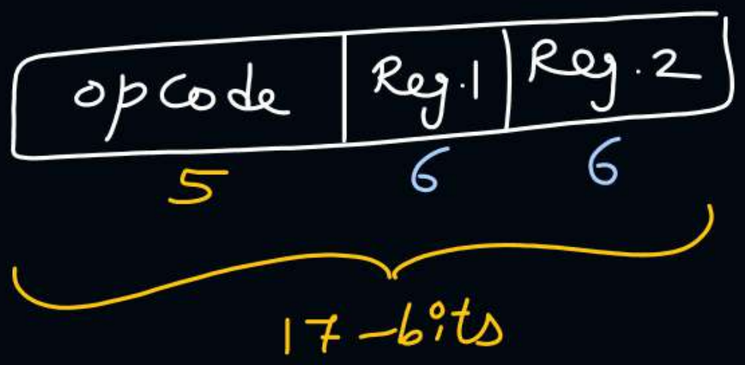
Total no. of instns = 10 + 12 + 4 = 26 instns

⇓

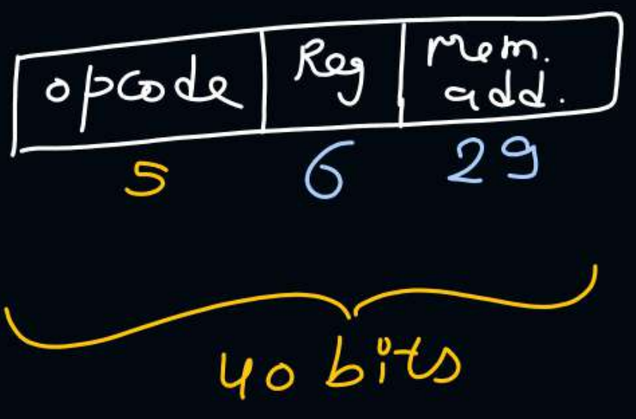
opcode = 5-bits

Maximum and Minimum instruction length are?

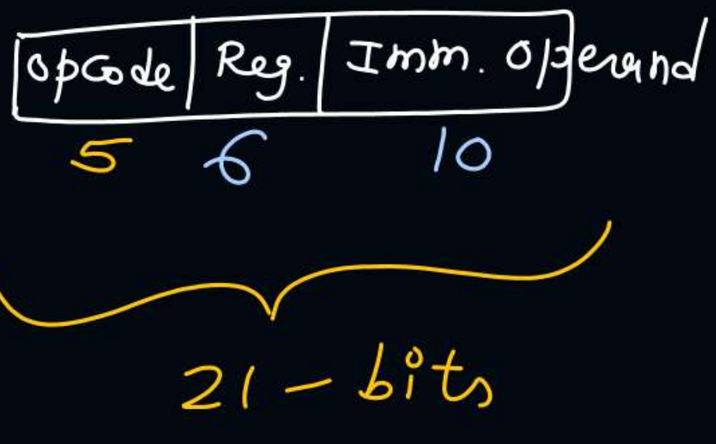
Reg-operand inst^{n}



Mem.-operand inst^{n}



Imm. operand inst^{n}



[NAT]
GATE

Reg-mem. based architecture ⇒ second operand always from Reg. for ALU operation

2 - add. + inst^n + 2nd operand is destination

#Q. In a simplified computer the instructions are:

OP R_i, R_j	- Performs R_i Op R_j and stores the result in R_j
OP m, R_i	- Performs val Op R_i and stores the result in R_i val denotes the content of memory location m
MOV m, R_i	- Moves the content of memory location m to register R_i
MOV R_i, m	- Moves the content of register R_i to memory location m

The computer has only two registers and OP is either ADD or SUB. Consider the following basic block:

R1 t1 = a + b
~~R2~~ t2 = c + d
~~R1~~ t3 = e – t2
R2 t4 = t1 – t3

(a, b, c, d, e) ⇒ operands in memory

t1, t2, t3, t4 ⇒ compiler temporaries

[NAT]

#Q. Assume that all operands are initially in memory. The final value of the computation should be in memory. What is the minimum number of MOV instructions in the code generated for this basic block?

Ans = 3

MOV b, R1	R1 ← b
ADD a, R1	R1 ← a + R1
MOV d, R2	R2 ← d
ADD c, R2	R2 ← c + R2
SUB e, R2	R2 ← e − R2
SUB R1, R2	R2 ← R1 − R2
MOV R2, mem.	mem. ← R2

[NAT]

Destinat^{n} ⇒ first operand
mem. operand ⇒ 2nd operand

#Q. In a simplified computer the instructions are:

OP R_i , R_j	- Performs R_i Op R_j and stores the result in R_i
OP R_i, m	- Performs R_i Op val and stores the result in R_i val denotes the content of memory location m
MOV m , R_i	- Moves the content of memory location m to register R_i
MOV R_i , m	- Moves the content of register R_i to memory location m

The computer has only two registers and OP is either ADD or SUB. Consider the following basic block:

R1 t1 = a + b
R2 t2 = c + d
t3 = e – t2
t4 = t1 – t3

[NAT]

#Q. Assume that all operands are initially in memory. The final value of the computation should be in memory. What is the minimum number of MOV instructions in the code generated for this basic block?

MOV a, R1	R1 ← a
ADD R1, b	R1 ← R1 + b
MOV c, R2	R2 ← c
ADD R2, d	R2 ← R2 + d
MOV R2, x	x ← R2
MOV e, R2	R2 ← e
SUB R2, x	R2 ← R2 − x
SUB R1, R2	R1 ← R1 − R2

MOV R1, mem mem ← R1

Ans = 5

Topic : Register Spill

If sufficient number of registers are not available in CPU then a Register value is Copied to memory for temporary basis, to carry out program execution.

[NAT]

Homework

#Q. Consider a register-memory architecture system (2-address instructions). For this system the following intermediate code is going to be converted in machine code. Minimum how many registers are required in system so that the code can run without register spill?

Consider first operand is always the register operand and it's the destination for operation too.

t1 = X + Y

t2 = t1 – Z

t3 = t1 + t2

t4 = M + t3

Assume X, Y, Z and M are memory operands

[NAT]

Homework

#Q. Consider a register-based architecture system (2-address instructions). For this system the following intermediate code is going to be converted in machine code. Minimum how many registers are required in system so that the code can run without register spill?

Consider first operand is always the register operand and it's the destination for operation too.

t1 = X + Y

t2 = t1 – Z

t3 = t1 + t2

t4 = M + t3

Assume X, Y, Z and M are memory operands

2 mins Summary

Topic Instructions

Topic Variable Length Instructions

Topic Register Spill

Happy Learning

THANK - YOU