Assignment-2

Name → Md. Farhan Haseen Prantor

ID → 21301536

Section → 12

Ans no: 1

for x10 – x17, there are 6 argument registers. If a function requires 10 parameters, then I will use stack for passing additional parameters because I have less registers for given parameters. I will pass 6 parameters into the x10 – x17 registers and rest of the 2 parameters in the stack.

Ans no: 2

From the opcode of a machine code, we can understand the basic operation of the instruction, and this abbriviation is its traditional name. The Opcode denotes the operation and format of an instruction, also distinguished by the values.

Ans no: 3

Based on the opcode & funct3 fields in the machine code, the machine will understand the size of the data.

LD x9, 10[x21]

In machine code, the opcode would be 0000011 for load operation & funct3 would be 011 to denote the data as double word (64 bits).

Ans no: 4

The register containing the address of the instruction in the program being executed is called program counter. Importance →

(1) After fetching an instruction, PC maintains the sequence by storing the pointer of next instruction.

(2) PC helps in enabling continious processing by modifying pointer of instruction.

Ans no: 5

Multiplying the immediate value by 2 adjusts a word offset to a byte offset, aligning it correctly with the memory addressing requirements. ~~By multiplying by 2,~~ we store half of the original value and while decoding, we first multiply by 2 for getting the actual value.

Ans no: 6

BEQ x0, x0, Label

In this instruction, both rs1 and rs2 fields would be always true because we store 0 in x0 register and it can not be changed. So the statement always returns true and jumps to the label. As we are intentionally making it to jump to anothe lable, it's an unconditional jump.

Ans no: 7

We need two different instruction for transferring control, because →

'JAL' is used for jumping to a function and return address. JAL x1, label transfers the control to the address labeled by 'label' & saves the return address in x1.

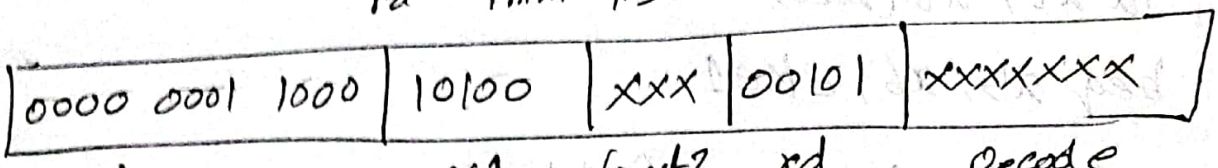
'JALR' is used for jumping to an address specified by a register, with an optional offset, saving the return address. JALR x0, label transfers control to the address computed by adding the offset and saves the return address in x0.

Ans no: 8(a)

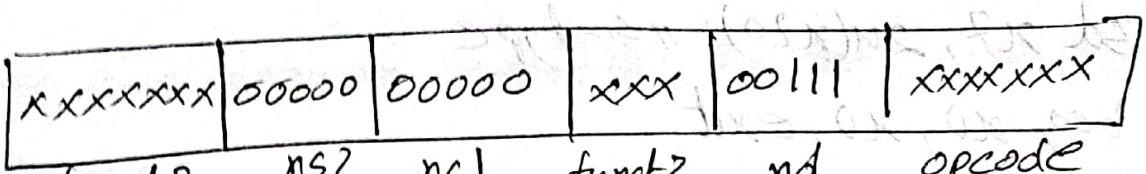
```
ld x5, 24(x20)  //I-type
ld x6, 48(x20)
beq x5, x6, else1
add x7, x0, x0  //R-type
bne x5, x7, else2
addi x7, x5, 2
sd x7, 24(x20)  //S-type
beq x0, x0, exit

else2:
srli x7, x6, 4
sd x7, 48(x20)
beq x0, x0, exit

else1:
slli x7, x6, 3
sd x7, 48(x20)
beq x0, x0, exit
exit:
```

Ans no: 8(b)

I-type ⇒ ld x5, 24(x20)
rd imm rs1



R-type ⇒ add x7, x0, x0
rd rs1 rs2



S-type ⇒ sd x7, 24(x20)
rs2 imm rs1