__/__/____

Tut-1

Combinatorial is another name for combinational

① Combinational circuit : outputs depends only on inputs. (no memory elements.)

Seq. circuit : Memory elements. (output depends on history of inputs.)

② GNU assembler. ('as'), g++ Compiler.

③ Algo → enhanced Java lang hierarchy.

HLL —Java Compile→ Java byte code

As. lang ← Java virtual machine.

Mach lang (responsible for doing mach dependent things)

D.L

④ Floppy disks.

⑥ If parallel tasks then → octa core ; {depends on clockspeed, Power consump etc}

⑨ MIPS ⇒ microprocessor without interlocked pipeline stages.

⑩ Networking devices & Automotive devices.

Q.7.

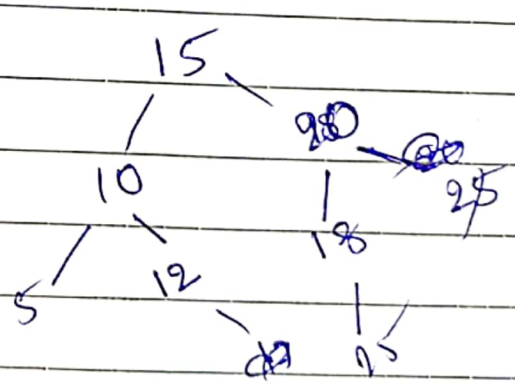
```
        addi  $s0, $zero, 100
FUNC:   addi  $s2, $zero, 0
        slt   $t0, $s2, $s0
        beq   $t0, $zero, EXIT
        sll   $t0, $s2, 4
        add   $s3  $t0, $s1
        lui   $t0, 1        (2^16)
        addi  $t0, $t0, 3   (2^16+3)  (65536+3)
        sw    0($s3), $t0   → x
        sw    4($s3), $t0   } a (2^32)(2^16+3)
        sw    8($s3), $0
        addi  $t0, $s2, 3
        sw    12($s3), $t0
        addi  $s2, $s2, 1
        j     FUNC
EXIT:
```

s3

char * y; // 32 bits.
};

align &M(i) → ptr
(16 bytes for each M)

Note big endian - higher order 4 bytes is at lower address



$S0 - save
$S1 → i
$S2 → k

Q.8

WHILE:

~~addi $s2, $zero, k~~

addi

muli $t0, $s1, 4 replace with sll $t0, $s1, 2

(a) add $s3, $t0, $s0

lw $t0, 0($s3)

beq $t0, $s2, NEXT

j EXIT

NEXT:

addi $s1, $s1, 1

j WHILE

EXIT:

(b)

bne $t0, $s2, EXIT

addi $s1, $s1, 1

j WHILE

EXIT:

This is the expected solution for (a)
For (b)
you need to implement like a do-while loop where condition is checked after body
However, since it is actually a while loop, you need to jump past the body straight to loop check (this jump will be executed only once)

(c) (a) = (b) + 1

(d) (a) → 70+6 (b) → 70+5