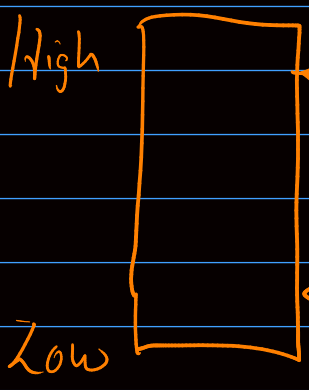
```
addi $sp, $sp-8      (allocate 8 bytes)
sw   $ra, 4($sp)
sw   $a0, 0($sp)

slti $t0, $a0, 1
beq  $t0, $0, L1
nop  (for pipelined) (Works without that as well)
```



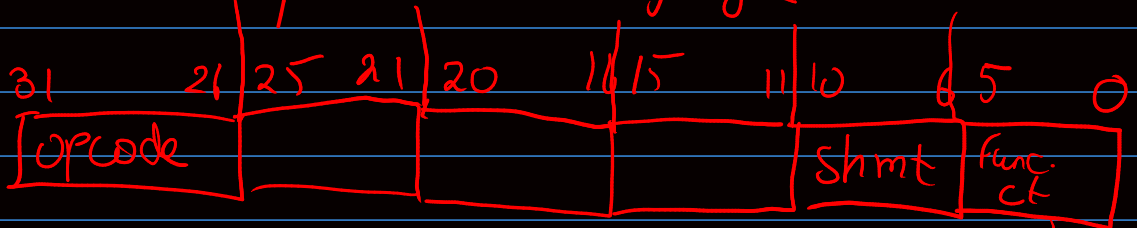
$fp - Points to the beginning of stack (frame), area of stack is called frame

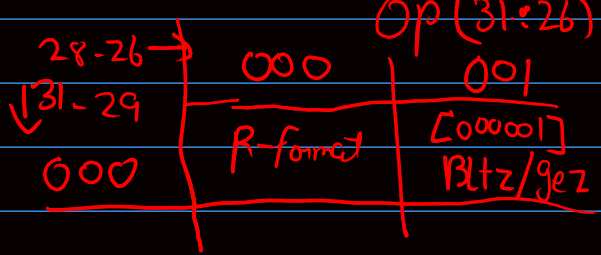
better to use this

$sp => end of stack

↳ Usage is optional (may not be efficient if no local vars)

Decoding Machine language





instruction format in Appendix A

also in green sheet

Binary Reph

0101100 ··· 00111

MSB ↩ Unsigned ↪ LSB

Sign Magnitude }
One's Complement } redundant '0' (ambiguous), have balance

two's complement } no balance between the -ve & +ve
No redundant '0'

Can we get balance & non redundancy (unambiguity)

no.

Overflows

- Overflow in arithmetic operation is that when the result has more number of digits than the operands.
- Overflow may occur only when two positive or negative numbers are added.
- For example: 5+3, -5-3?
- For signed numbers, overflow can be detected by performing an XOR of MSB of the result and carry out of the MSB.
- For example: -5-3?

Operation	Operand A	Operand B	Result Indicating Overflow
A + B	≥ 0	≥ 0	< 0
A + B	< 0	< 0	≥ 0
A – B	≥ 0	< 0	< 0
A – B	< 0	≥ 0	≥ 0

- Consider the operations A + B, and A – B
  - *can overflow occur if B is 0 ?*
  - *can overflow occur if A is 0 ?*

+3 + 0 : no overflow
-4 + 0 : no overflow
3 - 0 : no overflow
-4 - 0 : no overflow

0 + 3, 0 + (-4), 0 - (+3): no overflow

0 - (-4): Overflow (no representation of +4 in 3 bits)

Exception: Control jumps to predefined address for exception
~~Int addr is saved~~ addr is saved for return jic

add addi sub (overflow)
addu addiu subu (no overflow)