

Logic to check Least Significant Bit (LSB) of a number

We use Bitwise AND `&` operator to check status of any bit. Bitwise AND `&` operator evaluate each bit of the resultant value as `1` , if corresponding bits of both operands are `1` .

To check LSB of a number we need to perform bitwise ANDing. The bitwise AND operation `number & 1` will evaluate to `1` if LSB of number is set i.e. `1` otherwise evaluates to `0` .

12	←	In decimal	→	15
0000 1100	←	In binary	→	0000 1111
0000 0001	←	Binary of 1	→	0000 0001
<u>0000 0000</u>	←	Bitwise AND operation with 1	→	<u>0000 0001</u>

As you can see in above image `12 & 1` evaluate to `0` . Since, LSB of `12` is `0` . Whereas, `15 & 1` evaluate to `1` since LSB of `15` is `1` .

Lsb = least significant bit(binary olarak yazınca en sağdaki değer)

7 kez sağa kaydırma = 1 kez sola kaydırma.

Kırmızı renk = while loop(döngü)

Mavi renk = break; (döngüden çıkma)

- Write a program in the machine language of Appendix C that reverses the contents of the memory cell at address 8C. (That is, the final bit pattern at address 8C when read from left to right should agree with the original pattern when read from right to left.)

<u>Address</u>	<u>Contents</u>
00,01	2008 Register 0'a 08 değerini kaydet.(for i<8)
02,03	2101 Register 1'e 01 değerini kaydet.
04,05	2200 Register 2'ye 00 değerini kaydet.(i=0)
06,07	2300 Register 3'e 00 değerini kaydet.
08,09	148C 8C adresindeki değeri register 4'e kaydet.
0A,0B	8541 R4 ile R1'deki değerleri AND'le ve R5'e kaydet(yani lsb'yi al)
0C,0D	7335 R3 ve R5'teki değerleri ORla ve R3'e kaydet.
0E,0F	6212 R1 ve R2'deki değerleri topla ve R2'ye kaydet.(i++)
10,11	B218 Eğer R0==R2 ise 18'e atla. (if i==8) Değilse atlama.
12,13	A401 R4'teki değeri 1 kez sağa kaydır. (2^1 'e böl)
14,15	A307 R3'teki değeri 1 kez sola kaydır. (2^1 'le çarp)
16,17	B00A Eğer R0==R0'sa(kesin öyle) 0A'ya atla.
18,19	338C R3'teki değeri 8C adresine sakla.
1A,1B	C000 Bitir.