PAGE NO.:
DATE: / /

CO - OBA - 2

Venkatesh G D
2GI19CS175

1.) Booth's Algorithm:

i.) 28 x (-4)

28 = 011100

4 = 000100

∴ 2's complement of 4 = 111100

Booth recoding of -4 = 000-100

Calculation:

```
                  x 0 1 1 1 0 0
                    0 0 0 -1 0 0
  ---------------------------------
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  1 1 1 1 1 0 0 1 0 0
  0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0
  ---------------------------------
  1 1 1 1 1 0 0 1 0 0 0   (-112)
```

⇒ By Booth's algorithm product of 28 & -4 is -112

1.) ii.) -32×4

$\longrightarrow$ $32 = 0100000$

$4 = 0000100$

2's complement of

$(-32)\ 32 = 1100000$

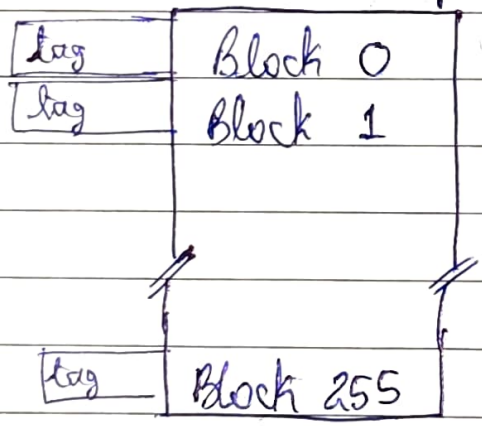
$(-4)\ 4 = 1111100$

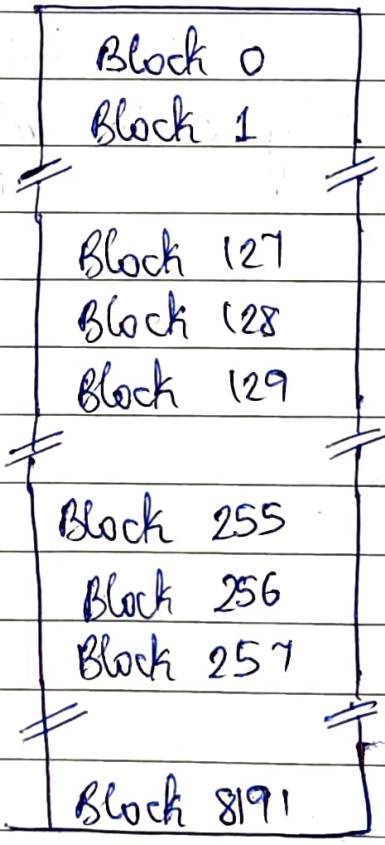
$$
\begin{array}{r}
1100000 \\
\times\ 0000\text{-}100 \\
\hline
0000000\ 0000000 \\
0000000\ 000000 \\
0000010\ 00000 \\
000000\ 0000 \\
00000\ 000 \\
0000\ 00 \\
000\ 0 \\
00\ 0 \\
\hline
0000\ 1000\ 0000 \quad (128)
\end{array}
$$

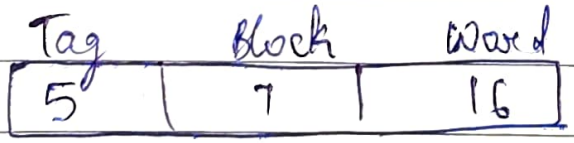
$\Rightarrow$ $\therefore$ By Booth's algorithm $-32 \times 4 = 128$

---

* Direct Mapping.





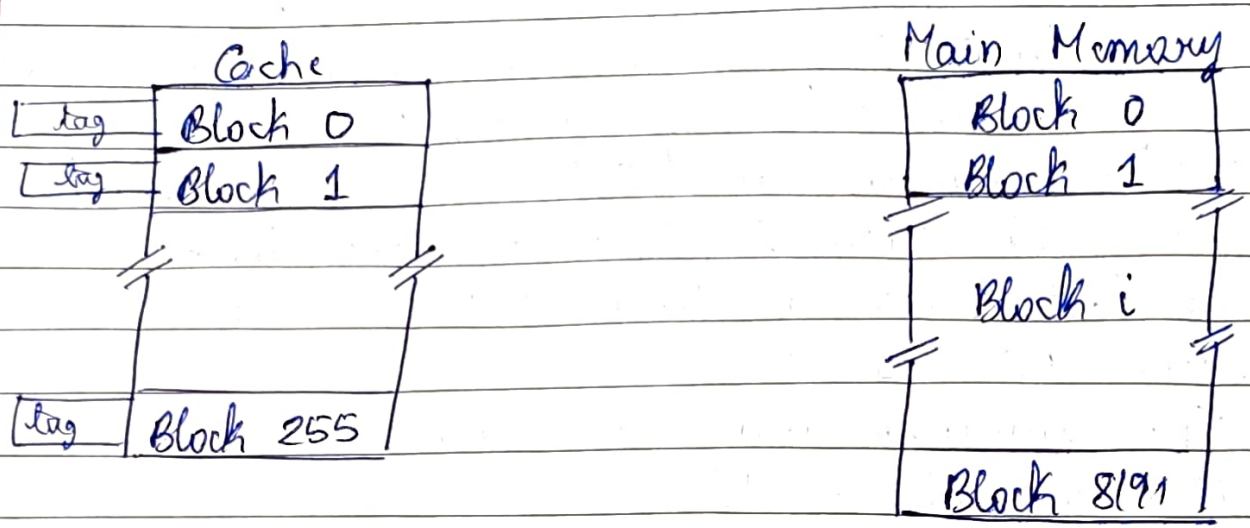


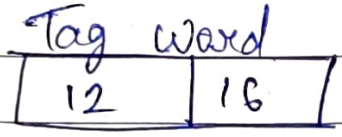
- Main Memory Address

2.) → The block -j of the main memory maps onto block j modulo -256 of cache.
→ When memory blocks 0, 128, & 256 are loaded into cache, the block is stored in cache block 0
Similarly, memory blocks 1, 129, 257 are stored in cache block 1.
→ The contention may arise when
i) When cache is full
ii) When more than one memory-block is mapped onto a given cache-block position.
→ The contention is resolved by allowing the new blocks to ~~overtime~~ overwrite the currently ~~res~~ resident-block
→ The memory address is divided into 3 fields
i) Low order 4 bit field - Selects one of 16 words in block
ii) 7 bits determine the cache-position in which new block must be stored.
iii.) 5 bit tag field - It is stored in 5 tag bits associated with cache location.

* Associate Mapping
→ The memory block can be placed into any cache block position
→ 12 tags bits will identify a memory-block when it is resolved in cache.
→ Tag-bits of an address received from processor are compared to the tag-bits
→ It gives complete freedom in choosing the cache-location.

2.)





→ A new block that has to be brought into the cache has to replace an existing block if cache is full.

→ The memory has to determine ~~exactly~~ whether a given block is in cache.