## Hardware Architecture for Deep Learning - CS6490. Spring 2023-24.
## Dept. of CSE, IIT Hyderabad

**Quiz-3**

**Total marks: 10** **Time: 12 minutes**

**Name:** Rahul Vigneswaran K

7

**Roll number:** CS23MTECH02002

---

1. In CNN accelerators, the area occupied by compute units is significantly larger than that of local storage (buffers or SRAMs): [1]
   a. True
   [✓ b. False]

1

2. Huffman encoding is a lossless compression scheme: [1]
   [✓ a. True]
   b. False

1

3. Briefly explain how/why loop tiling improves the execution time. [2]

→ Loop tiling would reuse that specific tile at hand :

→ This way we process those tiles individually instead of processing as whole .

~~→~~

But why is processing tile-wise more efficient?

0

4. Briefly explain the importance of using a dual-port SRAM in accelerators. [2]

→ When there are 2 port, we can use the 1 port to write things onto the SRAM while reading out through another.

→ So we do have to wait for the reading to be done in order to start writing. Both can happen at same time, saving wasted waiting time .

2

5. Consider the 4x4 matrix as shown below and assume that we have 4 bins 0, 1, 2, 3 with centroids as -1.0, 0.0, 1.0, and 2.0, respectively. Encode the matrix using these bins in a similar manner as discussed in the Deep Compression paper. [2]

| | | | |
|---|---|---|---|
| 2.10<br>3 | -1.10<br>0 | 2.09<br>3 | 1.05<br>2 |
| 0.07<br>1 | -1.02<br>3 | 1.31<br>2 | -0.01<br>1 |
| 2.03<br>3 | -0.06<br>1 | 1.02<br>2 | 1.99<br>3 |
| -1.01<br>0 | 2.01<br>3 | 1.12<br>2 | -0.98<br>0 |

~~3 0 3~~

3 0 3 2

1 3 2 1

3 1 2 3

0 3 2 0

| | |
|---|---|
| 0 | -1.0 |
| 1 | 0.0 |
| 2 | 1.0 |
| 3 | 2.0 |

1.5

Assuming that each value in the original matrix uses 32-bits, calculate the storage requirements (in bits) for the original and the encoded representation. [2]

Original : 16 × 32 - bit ✓

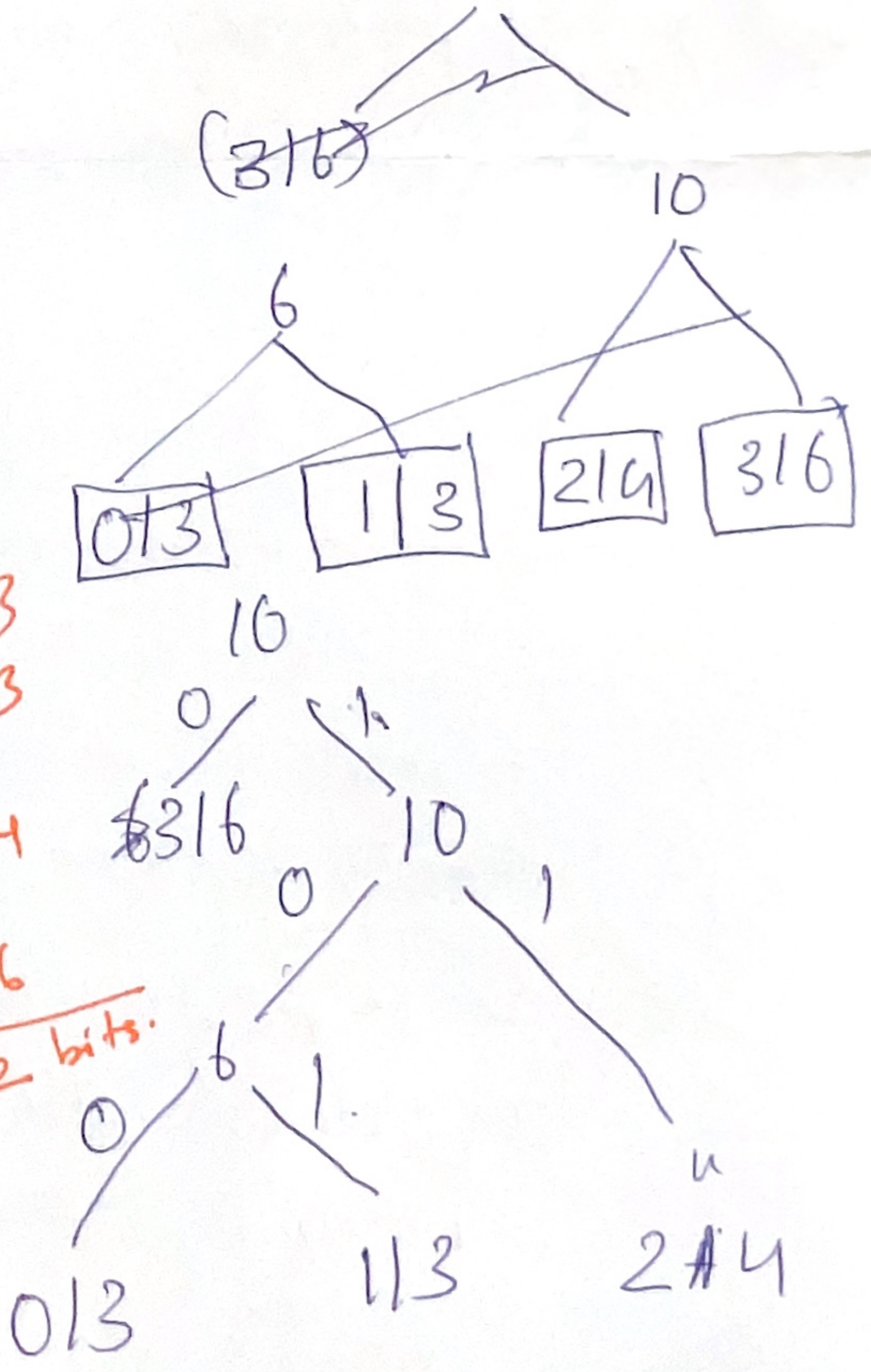
Encoded : 0 — 3times

1 — 3 times

2 — 4times

3 — 6 times

1.5



0 — 10 0 -3 bit ·x3

1 — 10 1 -3 bit x3

2 — 11 -2 bit x4

3 — 0 -1 bit x6

32 bits.

~~16x32~~

4.centroid x 32 bit + 3 bits Index

128 + 3 = 131 bits >> original