Assignment 1 – SystemC intro and L1 cache

Arun Jayabalan(368048), Chethan Shettar(368083), Varun Gowtham(368053)

Task:

In this task you must build a single 32kB 8-way set-associative L1 D-cache with a 32-Byte line size. The simulator must be able to be driven with the single processor trace files. Assume a Memory latency of 100 cycles, and single cycle cache latency. Use a least-recently-used, write-back replacement strategy.

The address from the processor is of 32 bits.
Since the line size is 32-Byte, offset = 2^5 = 32 Bytes => 5 bits for offset (bit 0 to bit 4)
Cache block = 32KB/32Bytes = 1KB Cache blocks
Number of sets = 1024/8 = 128 =2^7 => 7 bits for index (bit 5 to bit 11)
Tag field is the remaining 20 bits.
Pseudo LRU replacement bits = 7(N way -1)

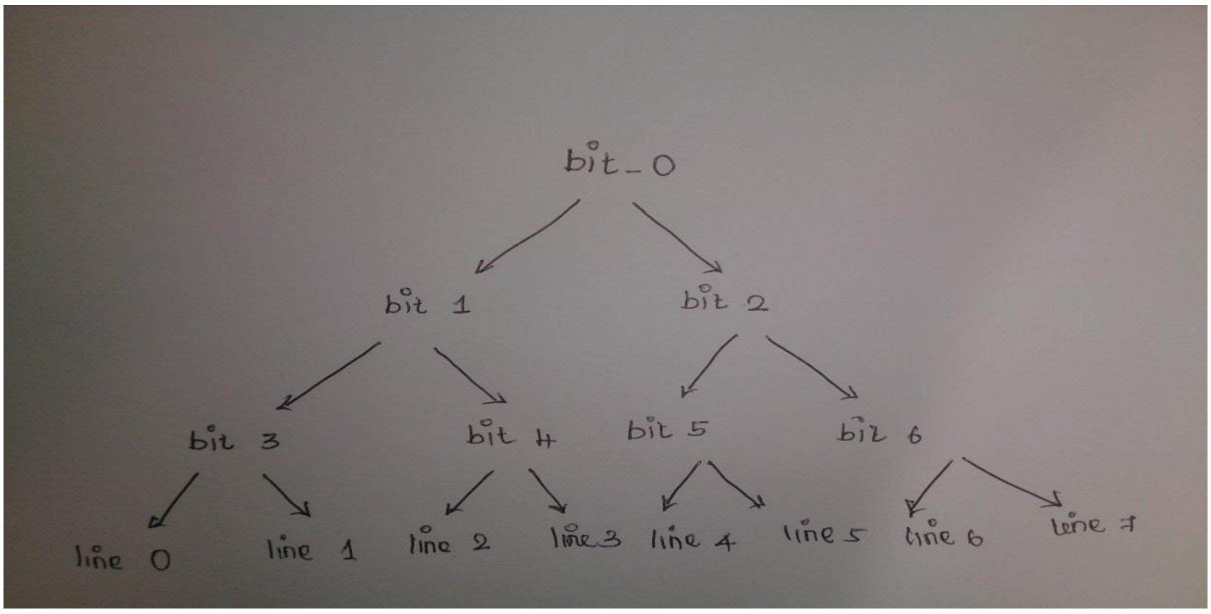
```
#define TAG_CHECK 0xFFFFF000
#define SET_CHECK 0x00000FE0
#define OFF_CHECK 0x0000001F

unsigned int tag = (addr & TAG_CHECK) >> 12; //addr taken from the trace file
unsigned int set = (addr & SET_CHECK) >> 5;
unsigned int word = (addr & OFF_CHECK) >> 2;
```

For the replacement strategy, pseudo LRU algorithm has been implemented. The binary tree structure and the state table for the algorithm are given below. Bit mask is used to get the next state from the present state in a particular set.

State table:

Way	Present state	Bit mask	Next State
Way_0	xxx0x00	0x0B	---1-11
Way_1	xxx1x00	0x0B	---0-11
Way_2	xx0xx10	0x13	--1--01
Way_3	xx1xx10	0x13	--0--01
Way_4	x0xx0x1	0x25	-1--1-0
Way_5	x1xx0x1	0x25	-0--1-0
Way_6	0xxx1x1	0x45	1---0-0
Way_7	1xxx1x1	0x45	0---0-0

State diagram:



Results:

Trace	CPU	Reads	RHit	RMiss	Writes	Whit	Wmiss	HitRate
DBG	0	40	19	21	60	26	34	45
RND	0	33031	18659	14372	32505	18306	14199	56.4
FFT	0	86298	7652	78646	43195	10882	32313	14.31

Observations:

- Calculating the tag field and set field correctly from the address is the main task to get the appropriate hit rate for the architecture
- The hit rate gets messed up unless the LRU implementation is correct
- For pseudo LRU implementation, next state has to be updated every time irrespective of read/write hit/miss.