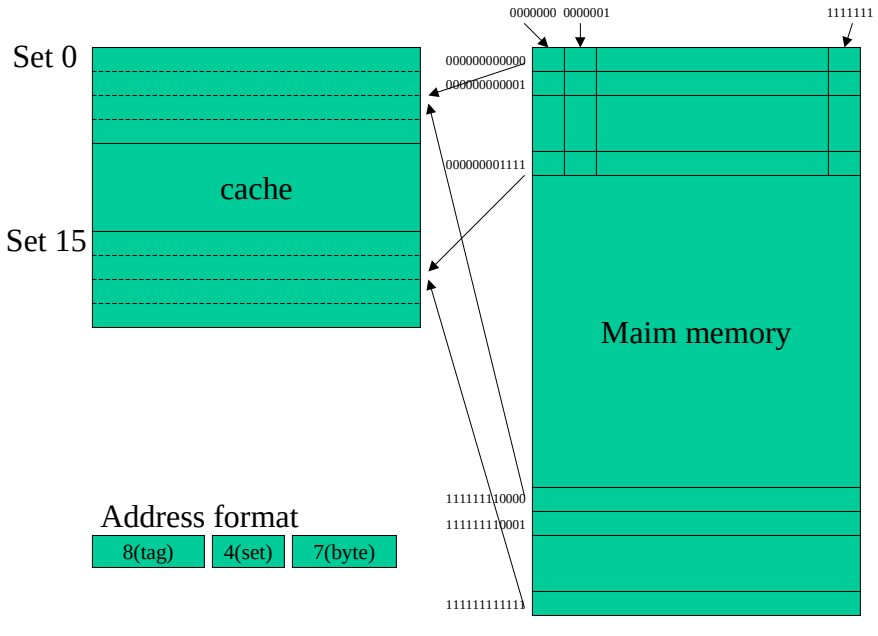
For this set associative cache, cache is divided into 16 sets, each set contains 4 lines, the size of each line of the cache is the size of each block of the memory, 128 word in this case. A given block maps to any line in its mapping set.

The memory address space is 4K x 128 words

Therefore, the cache scheme needs
7 least significant address bits to decide the unique unit inside the block;
4 address bits to decide the set address
the rest 8 most significant address bits will be used as tag to identify whether the copy of the memory is in cache or not when each time the memory address is applied to access the cache.



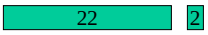
The size of each memory block is 4 Bytes, therefore, 2 least significant bits are saved for within block addressing.
The 64KB cache is divided into 16K lines. The 16MB memory has 24 bits of address

(a) for the cache with direct mapping:
word field has 2 bits; the line field has 14 bits; the tag field has 8 bits

address	tag(8b)	line(14b)	word(byte)(2b)
111111	11	0444	1
666666	66	1999	2
BBBBBB	BB	2EEE	3

(b) for the associative cache
word field has 2 bits; the tag field has 22 bits

address	tag(22b)	word(byte)(2b)
111111	044444	1
666666	199999	2
BBBBBB	2EEEEE	3



(c) for two-way set associative cache

address	tag(9b)	set(13b)	word(byte)(2b)
111111	022	0444	1
666666	0CC	1999	2
BBBBBB	177	0EEE	3

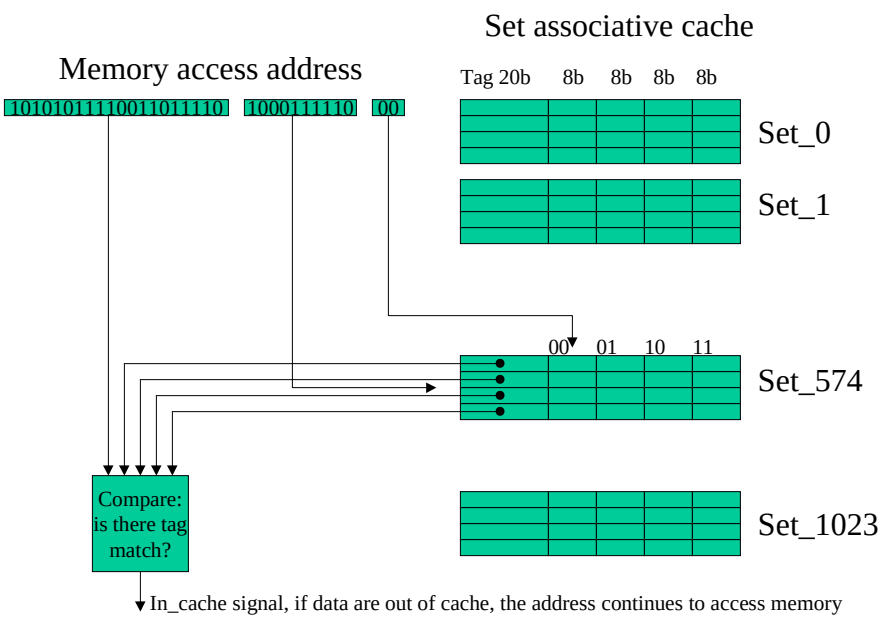
The block size of the memory is 8 bytes, the cache consists 32 line with direct mapping

(a) the total memory can be stored in cache equals to the cache size, 32*8 Bytes

(b) the 16 bit memory address will be divided into 3-bit byte field, 5-bit line field, and the rest 8 bit tag field

(c)
for memory address 0001 0001 0001 1011, the line index is 00011
for memory address 1100 0011 0011 0100, the line index is 00110
for memory address 1101 0000 0001 1101, the line index is 00011
for memory address 1010 1010 1010 1010, the line index is 10101

(d) all the content in the same line with the same tag will be stored along with each other they are
0001 1010 0001 1000
0001 1010 0001 1001
0001 1010 0001 1011
0001 1010 0001 1100
0001 1010 0001 1101
0001 1010 0001 1110
0001 1010 0001 1111



Main memory consists of 512 blocks of 64 word. Cache consists 0f 16 sets, each set consists of 4 lines, each line consists of 64 words. Here, a word(16bit) is the addressable unit.
Suppose the access time is considered only by each block unit access.

Therefore, one repetition involves 68 times of block access.
During first access iteration, all accesses are missed in cache, but after the first iteration, the data from the memory has been loaded into the cache, while the first 4 sets have been swap out once by LRU algorithm, the reset 12 sets are disturbed. For the consecutive repetition access, the first 4 sets of the cache are always missed because of the update, while the reset 12 sets are always hit.

Suppose the cache access time is Tc, while memory access time is Tm

the total accessing time with such cache is 68*(Tm+Tc) + 9*20*(Tm+Tc) + 9*48*Tc

the total accessing time with cache support is 10*68*Tm

the improvement is (68*10*10*Tc)/(68*11*Tc+9*20*11*Tc+9*48*Tc)=2.15

(Tm=10*Tc)