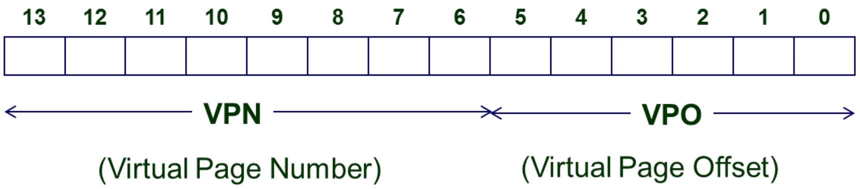
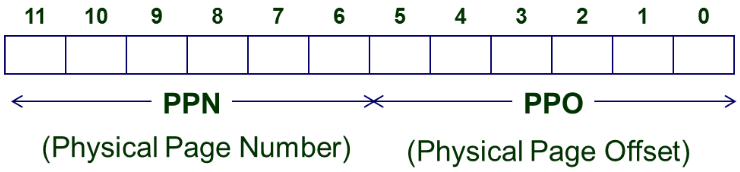
CSE 560 – Practice Problem Set 7 Solution

1. Consider the simple virtual memory system described below.
   - 14-bit virtual addresses
   - 12-bit physical addresses
   - Page size = 64 bytes

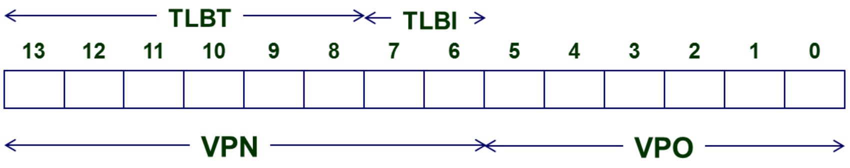




The first 16 entries in the virtual page table are shown below:

VPN	PPN	Valid	VPN	PPN	Valid
00	28	1	08	13	1
01	–	0	09	17	1
02	33	1	0A	09	1
03	02	1	0B	–	0
04	–	0	0C	–	0
05	16	1	0D	2D	1
06	–	0	0E	11	1
07	–	0	0F	0D	1

The TLB has 16 entries and is 4-way associative. We can understand access to the TLB by considering virtual addresses as follows:

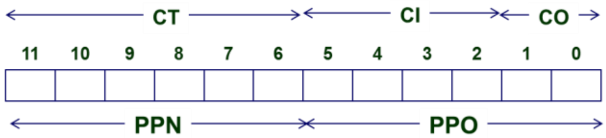


Here the TLBT is the tag bits and the TLBI is the index into the TLB.

The contents of the TLB are shown next.

Set	Tag	PPN	Valid	Tag	PPN	Valid	Tag	PPN	Valid	Tag	PPN	Valid
0	03	–	0	09	0D	1	00	–	0	07	02	1
1	03	2D	1	02	–	0	04	–	0	0A	–	0
2	02	–	0	08	–	0	06	–	0	03	–	0
3	07	–	0	03	0D	1	0A	34	1	02	–	0

The physically addressed cache is direct mapped. It has 16 lines (blocks) and a 4-byte line size.

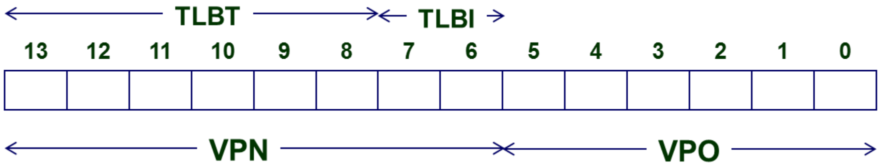


Here, CT is the cache tag, CI is the index, and CO is the offset.

The contents of the cache are:

Idx	Tag	Valid	B0	B1	B2	B3	Idx	Tag	Valid	B0	B1	B2	B3
0	19	1	99	11	23	11	8	24	1	3A	00	51	89
1	15	0	–	–	–	–	9	2D	0	–	–	–	–
2	1B	1	00	02	04	08	A	2D	1	93	15	DA	3B
3	36	0	–	–	–	–	B	0B	0	–	–	–	–
4	32	1	43	6D	8F	09	C	12	0	–	–	–	–
5	0D	1	36	72	F0	1D	D	16	1	04	96	34	15
6	31	0	–	–	–	–	E	13	1	83	77	1B	D3
7	16	1	11	C2	DF	03	F	14	0	–	–	–	–

(a) For virtual address 0x03D4, fill in the bits of the virtual address below (yes, in binary) and answer the associated questions.



0 0 0 0 1 1 1 1 0 1 0 1 0 0

What is the VPN (in hex)? 0x0f

What is the VPO (in hex)? 0x14

What is the TLBI (in hex)? 0x3

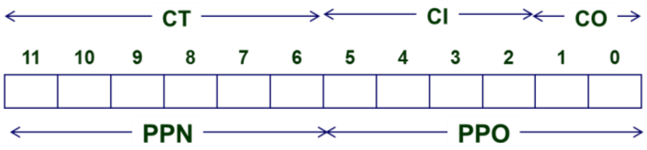
What is the TLBT (in hex)? 0x03

Does an access to this address result in a TLB hit? yes

Does an access to this address result in a page fault? no

What is the PPN (in hex)? 0x354 (see below)

If known, fill in the bits of the physical address below (yes, in binary) and answer the associated questions.



0 0 1 1 0 1 0 1 0 1 0 0

What is the cache offset (in hex)? 0x0

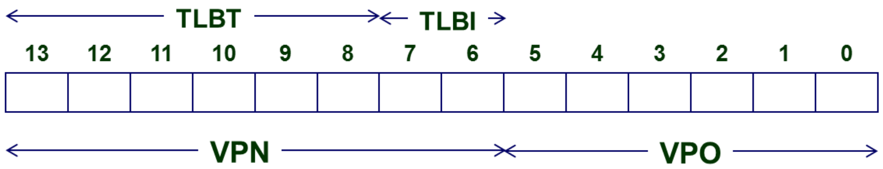
What is the cache index (in hex)? 0x5

What is the cache tag (in hex)? 0x0d

Does an access to this address result in a cache hit? yes

If so, what is the byte value? 0x36

(b) Repeat part (a) for virtual address 0x0B8F.



0 0 1 0 1 1 1 0 0 0 1 1 1 1

VPN is 0x2e

VPO is 0x0f

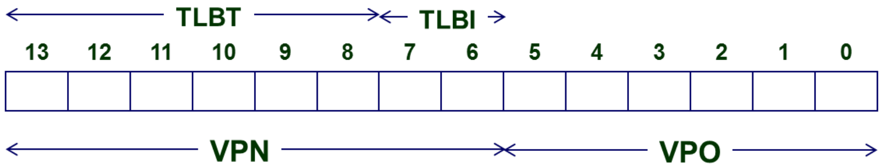
TLBI is 0x2

TLBT is 0x0b

TLB hit? no

page fault? Can’t tell from given page table (it only shows first 16 pages).

(c) Repeat part (a) for virtual address 0x0040.



0 0 0 0 0 0 0 0 0 0 0 1 0 0

VPN is 0x00

VPO is 0x04

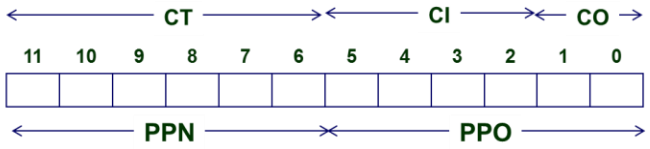
TLBI is 0x0

TLBT is 0x00

TLB hit? no

page fault? No

PPN is 0xc04



1 0 1 0 0 0 0 0 0 1 0 0

Cache offset is 0x0

Cache index is 0x1

Cache tag is 0x28

Cache hit? No

2. Starting from the standard equation for memory access time with a cache:

$$t_{avg} = t_{hit} + \%miss \times t_{miss}$$

Expand the equation to include the effects of virtual memory, including a TLB, and both an L1 and L2 cache. Be careful with the definitions of any variables you introduce.

Define the following variables to have the given definitions. The first is what we are trying to compute, and the rest are all inputs (i.e,. given).

Variable	Definition
t_{avg}	Average memory access time
$t_{hit-TLB}$	Time for a hit in the TLB
t_{hit-L1}	Time for a hit in the L1 cache
t_{hit-L2}	Time for a hit in the L2 cache
t_{MM}	Main memory access time
t_D	Secondary store (disk) access time
$\%miss_{TLB}$	TLB miss rate
$\%miss_{L1}$	L1 miss rate
$\%miss_{L2}$	L2 miss rate
$\%fault$	Page fault rate

We will also use the following intermediate variables:

Variable	Definition
$t_{miss-TLB}$	Average TLB miss time
$t_{miss-L1}$	Average L1 miss time
$t_{miss-L2}$	Average L2 miss time
t_{fault}	Average page fault time

Working from the processor's access down the hierarchy,

$$t_{avg} = \max(t_{hit-TLB} + \%miss_{TLB} \times t_{miss-TLB}, t_{hit-L1} + \%miss_{L1} \times t_{miss-L1})$$

The above equation implicitly assumes that TLB access and L1 access happen concurrently. Depending upon how the memory system is constructed, an alternate expression would be

$$t_{avg} = \max(t_{hit-TLB}, t_{hit-L1}) + \%miss_{TLB} \times t_{miss-TLB} + \%miss_{L1} \times t_{miss-L1}$$

The formula for average L1 miss time should be familiar:

$$t_{miss-L1} = t_{hit-L2} + \%miss_{L2} \times t_{miss-L2}$$

$$t_{miss-L2} = t_{MM}$$

We next need to consider when we have a miss in the TLB

$$t_{miss-TLB} = t_{MM} + \%fault \times t_{fault}$$

$$t_{fault} = t_D$$

The above assumes a single page table access. If a multi-level page table is used, the t_{MM} in the above expression for $t_{miss-TLB}$ would need to be changed to $2t_{MM}$.

The above expressions assume a relatively straightforward memory hierarchy.