154B Discussion 8

March 1st, 2023

Outline

- Assignment 4
- VM Quiz

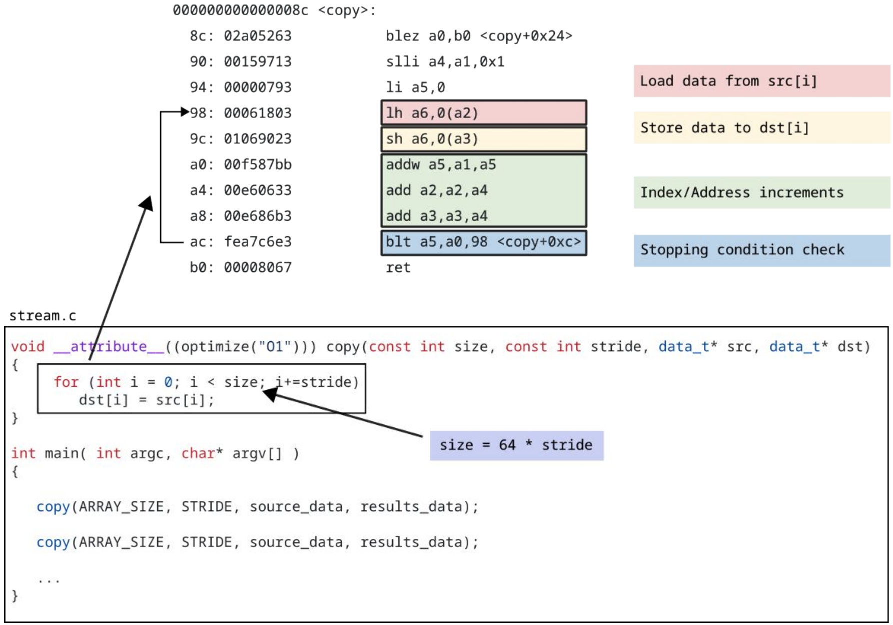
Assignment 4

- Question 1: Use the single-cycle cpu

```
runMain dinocpu.simulate <binary_name> single-cycle
```

Assignment 4

- Question 3: effective data bandwidth
    - Does not count bandwidth used for bringing in instructions.



```
000000000000008c <copy>:
  8c: 02a05263          blez a0,b0 <copy+0x24>
  90: 00159713          slli a4,a1,0x1
  94: 00000793          li a5,0
  98: 00061803          lh a6,0(a2)
  9c: 01069023          sh a6,0(a3)
  a0: 00f587bb          addw a5,a1,a5
  a4: 00e60633          add a2,a2,a4
  a8: 00e686b3          add a3,a3,a4
  ac: fea7c6e3          blt a5,a0,98 <copy+0xc>
  b0: 00008067          ret
```

stream.c

```
void __attribute__((optimize("O1"))) copy(const int size, const int stride, data_t* src, data_t* dst)
{
   for (int i = 0; i < size; i+=stride)
      dst[i] = src[i];
}

int main( int argc, char* argv[] )
{

   copy(ARRAY_SIZE, STRIDE, source_data, results_data);

   copy(ARRAY_SIZE, STRIDE, source_data, results_data);

   ...
}
```

Assignment 4

Cond 1 : exmem_taken
2 : (rs1=rd || rs2=rd) & idex_memread
3 : exmem_meminstr & !dmem_good
4 : !imem_good

Case	C1	2	3	4	nextpc	ID	EX	MEM	WB
1	0	0	0	0	pc+4	~	~	~	~
2	0	0	1	0	pc stall	stall	stall	stall	flush
3	0	1	0	0	pc stall	stall	flush	~	~
4	0	1	1	0	pc stall	stall	stall	stall	flush
5	1	0	0	0	pc+4	x	flush	flush	~
6	1	1	0	0	pc+4	x	flush	flush	~
7	0	0	0	1	pc stall	flush	~	~	~
8	0	0	1	1	pc stall	stall	stall	stall	flush
9	0	1	0	1	pc stall	stall	flush	~	~
10	0	1	1	1	pc stall	stall	stall	stall	flush
11	1	0	0	1	taken pc	flush	flush	stall	flush
12	1	1	0	1	taken pc	flush	flush	stall	flush

Assignment 4

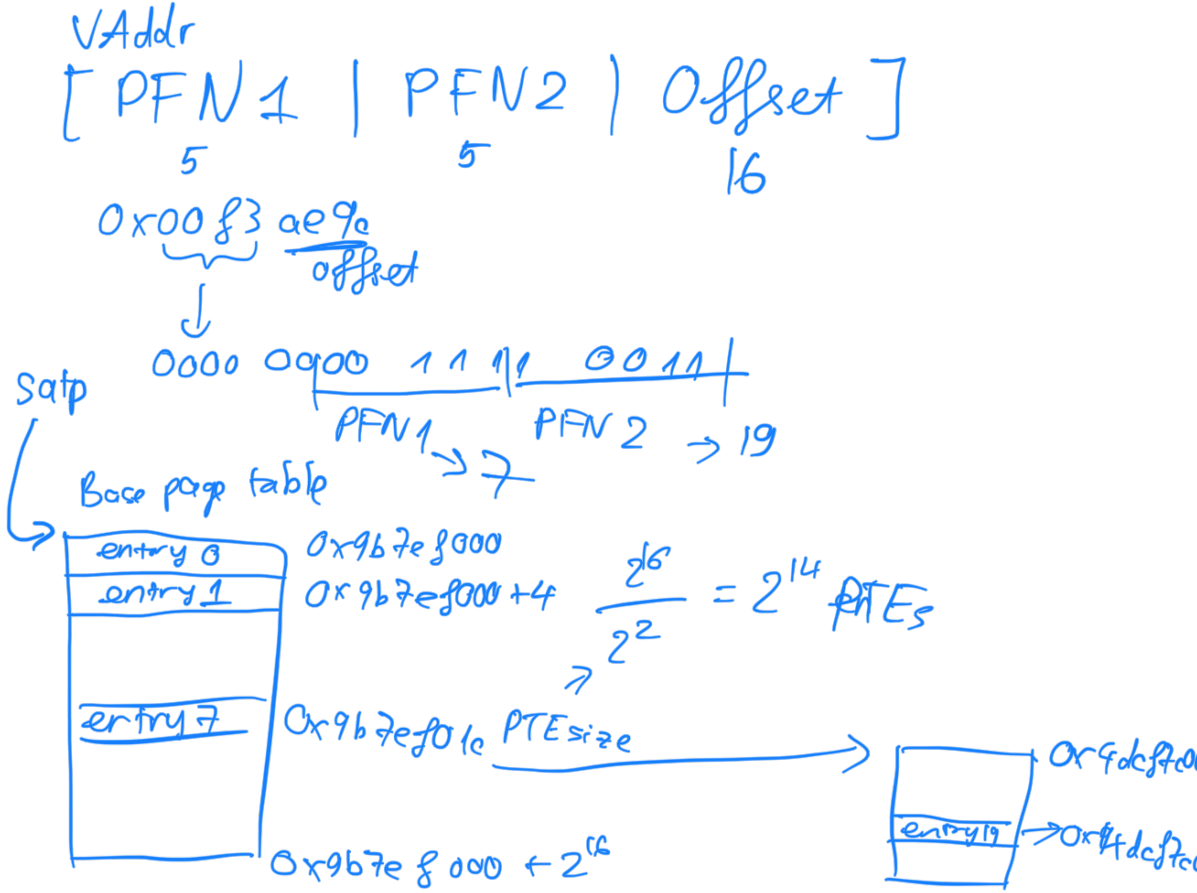
```
when (cond3) { // 2, 4, 8, 10
    pc stall
    ID, EX, MEM -> stall
    WB -> flush
}. elsewhen (cond1) { // 5, 6, 11, 12
    pc from taken -> cond4
    ID _flush -> cond4
    EX -flush -> true
    MEM_ stall -> cond4
    MEM_ flush -> ~ cond4
    WB- flush -> cond4
}
```

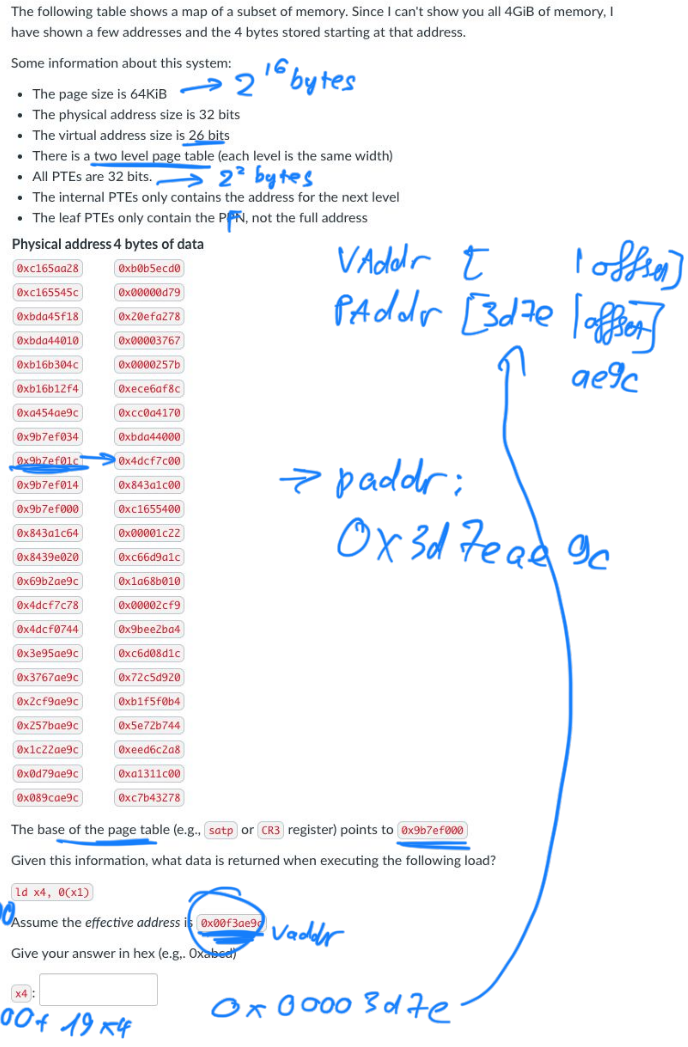
```
. elsewhen (cond2) { //3, 9
    pc stall
    ID -> stall
    EX -> flush
}
. elsewhen (cond4) { //7
    pc stall
    ID -> flush
}
```

Assignment 4

Assignment 4

VM Quiz





The following table shows a map of a subset of memory. Since I can't show you all 4GiB of memory, I have shown a few addresses and the 4 bytes stored starting at that address.

Some information about this system:

- The page size is 64KiB
- The physical address size is 32 bits
- The virtual address size is 26 bits
- There is a two level page table (each level is the same width)
- All PTEs are 32 bits.
- The internal PTEs only contains the address for the next level
- The leaf PTEs only contain the PFN, not the full address

Physical address	4 bytes of data
0xc165aa28	0xb0b5ecd0
0xc165545c	0x00000d79
0xbda45f18	0x20efa278
0xbda44010	0x00003767
0xb16b304c	0x0000257b
0xb16b12f4	0xece6af8c
0xa454ae9c	0xcc0a4170
0x9b7ef034	0xbda44000
0x9b7ef01c	0x4dcf7c00
0x9b7ef014	0x843a1c00
0x9b7ef000	0xc1655400
0x843a1c64	0x00001c22
0x8439e020	0xc66d9a1c
0x69b2ae9c	0x1a68b010
0x4dcf7c78	0x00002cf9
0x4dcf0744	0x9bee2ba4
0x3e95ae9c	0xc6d08d1c
0x3767ae9c	0x72c5d920
0x2cf9ae9c	0xb1f5f0b4
0x257bae9c	0x5e72b744
0x1c22ae9c	0xeed6c2a8
0x0d79ae9c	0xa1311c00
0x089cae9c	0xc7b43278

The base of the page table (e.g., `satp` or `CR3` register) points to `0x9b7ef000`

Given this information, what data is returned when executing the following load?

```
ld x4, 0(x1)
```

Assume the *effective address* is `0x00f3ae9c`

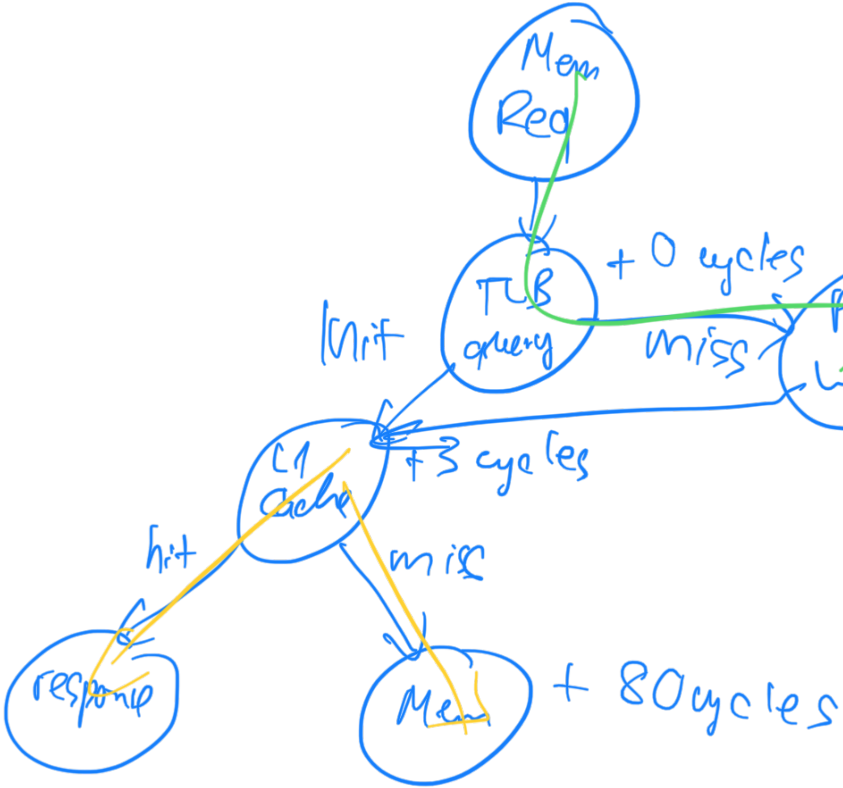
Give your answer in hex (e.g,. 0xabcd)

`x4`:

VM Quiz

Question 8

$$AMAT = TLB\_miss\ rate \times t_{Page\ walk} + L1\_latency + L1\ miss\ rate \times t_{mem\text{-}access}$$


Mem Req
TLB query
+ 0 cycles
hit
miss
Page Walk
+170 cycles
L1 cache
+3 cycles
hit
miss
response
Mem
+ 80 cycles