COSC 3380 — Operating Systems

Name: Michael Osei **Due Date:** January 30 2018
Homework: Number 1

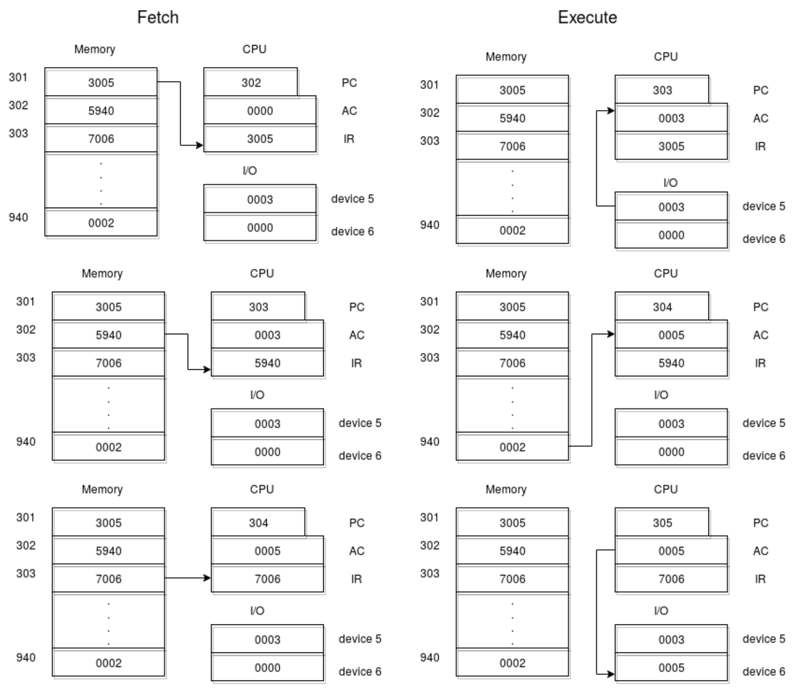
Question 1.1 Suppose the hypothetical processor of Figure 1.3 also has two I/O instructions:

- 0011 = Load AC from I/O
- 0111 = Store AC to I/O

In these cases, the 12-bit address identifies a particular external device. Show the program execution (using format of Figure 1.4) for the following program:

- Load AC from device 5.
- Add contents of memory location 940.
- Store AC to device 6.s

Assume that the next value retrieved from device 5 is 3 and that location 940 contains a value of 2.



Answer :

Question 1.7 In virtually all systems that include DMAA modules, DMA access to main memory is given higher priority than processor access to main memory. Why?

Answer: This is because DMA access is much more efficient for the processor than it having to use cycles to access main memory. With DMA the I/O operation is delegated to the DMA module so the actual transfer of memory blocks to and from main memory is handled by it, not the CPU. This means the CPU is only involved at the beginning and end of the I/O operation.

Question 1.10 Consider the following code:

```
for (i=0; i<20; i++){
        for(j=0; j<10; j++){
        a[i] = a[i] * j;
    }
}
```

1. Give one example of the spatial locality in the code.
2. Give one example of the temporal locality in the code.

Answer:

1. **Spatial locality**: The code above is accessing a 1d array of values in which the inner loop accesses values that are close to each other. This is assuming the language used does not support operator overloading where in the indexing operator may have been overloaded for a data structure like a linked list.
2. **Temporal locality**: Since the code is in a loop, the processor will cache those instructions used frequently (a[i] = a[i] ∗ j) in its instruction cache (if it has one).

Question 1.14 Suppose a stack is to be used by the processor to manage procedure calls and returns. Can the program counter be eliminated by using the top of the stack as the program counter?

Answer No, because of the following reasons

- The stack does not hold processor instructions. If it did then it would be no different from other parts of memory.
- Because some processors allow reentrant procedures the top of the stack cannot be used as the program counter.