Peter Rosser

374 0602

Pros028

Compsci 210 assignment 2, part B

1. An operation that would be easier would be subtracting one number from another, as trying to figure out the 2's compliment of a positive number can be mentally hard, where as i find it easy to calculate a positive number in binary.
2. A more difficult operation would be running a counter, as you have to have subtract a negative number to increment the counter, so it becomes harder to conceptualise, as you are subtracting but your counter has a positive value and is increasing.
3. I think this would be a bad idea. It is easier for humans, who are ultimately the programmers, to visualise 1 + 1 + 1 + 1 + 1 than it is to visualise 1 - - 1 - - 1 - - 1 - - 1, and would make it easier to create errors in the code
4. NOT flips the bits in the SR and puts them in the DR. If it used bit 5 to then designate whether it is flipping once or twice this would make no sense. Flipping all the bits twice leaves you with your original value, so there is no point in using the NOT command. And flipping all the bits once leaves you with NOT(original bits). So either you use the NOT command as it is, or you don't need to use it at all. I can't think of a purpose in calling a NOT(NOT(n)) command on your register n.