name:	Vince Rothenberg
course:	csci 10
assignment:	homework 6
prepared:	Thu, Mar 5, 2020 // 10:30 am

1. Use the Nucleo-64 Reference Manual to find the following Arduino -> GPIOA mappings: D13, D12, D11, D7.

D13 -> PA5

D12 -> PA6

D11 -> PA7

D7 -> PA8

GPIOA boundary addresses 0x4002 0000 - 0x4002 03FF

GPIOA_MODER Offset 0x00

2. What ARM instruction is used to enable (set to 1) bits?

The ORR instruction flips bits on. The following instructions ensures every bit in the 32 bit R0 register is on:

```
ORR R0, R0, 0xFFFFFFFF
```

3. What hexadecimal value is needed to enable the GPIOA pins identified above for output in GPIOA_MODER? Remember that GPIOA_MODER is 32-bits, each pin setting is 2-bits, and the setting for output is 01.

0x15400

0b0000_0000_0000_0001_0101_0100_0000_0000

4. What hexadecimal value is needed to enable the pins mapped to D13 and D11 in GPIOA_ODR? Remember that GPIOA_ODR is 16-bits, each pin setting is 1-bit.

0x2800 using ORR

5. What ARM instruction is used to clear (set to 0) bits?

The BIC instructions clears bits, setting them to 0.

```
BIC R0, R0, 0xFFFFFFFF
```

6. What hexadecimal value is needed to clear the pins mapped to D12 and D7 in GPIOA_ODR? Remember that GPIOA_ODR is 16-bits, each pin setting is 1-bit.

0x1080 using BIC

7. What ARM instruction is used to toggle (change 0s to 1s and 1s to 0s) bits?

EOR, the exclusive or instruction, toggles bits.

```
MOV   R0, 0xFF      # Set the register R0 to 0b1111_1111
EOR    R0, R0, 0x19 # Toggle with 0b0001_1001

Results in R0 = 0b1110_0110
```

8. What hexadecimal value is needed to toggle the pins mapped to D13, D12, D11 and D7 in GPIOA_ODR? Remember that GPIOA_ODR is 16-bits, each pin setting is 1-bit.

0x3880 using EOR

9. What is the range of values that can be loaded using the ARM MOV instruction? HINT: The Cortex-M4 on our board is ARMv7E-M.

The range of permitted values is 0-255 for encoding T1 and 0-65535 for encoding T3. When both 32-bit encodings are available for an instruction, encoding T2 is preferred to encoding T3 (if encoding T3 is required, use the MOVW syntax)

LDR and STR support 32 bit words. LDRD and STRD support 64 bit, by moving 2 words of 32 bits.

10. What is the purpose of the "LDR Rd, =const" pseudo-instruction?

The LDR Rd,=const pseudo-instruction can construct any 32-bit numeric constant in a single instruction. You can use this pseudo-instruction to generate constants that are out of range of the MOV and MVN instructions.