CS1021 Tutorial 2

Q1 The NXP LPC2468 has 512KiB of flash memory starting at address 0x00000000. What is the last address of the memory area (in hexadecimal)?

512KiB = 512 x 1024 = 524,288 bytes

Convert 524,288 to hexadecimal

16	524,288	
16	32,768	r 0
16	2,048	r 0
16	128	r 0
16	8	r 0
16	0	r 8

524,288 = 0x00080000

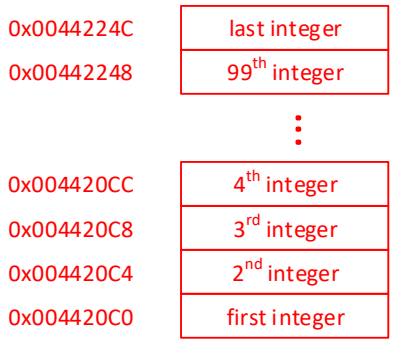
Last address of memory area is 0x00000000 + 0x00080000 – 1 = 0x0007FFFF

Need to subtract 1 because first byte at offset 0 (not 1).

Q2 The NXP LPC2468 has 64KiB of read write memory starting at address 0x40000000. What is the last address of the memory area (in hexadecimal)?

Q3 One hundred 8 bit unsigned integers are stored consecutively in memory starting at address 0x00002000. What is the address of the byte containing (i) the first integer (ii) the 22nd integer (iii) the 75th integer and (iv) the last integer?

Q4 One hundred 32 bit signed integers are stored consecutively in memory starting at address 0x004420C0. What is the address of the word containing (i) the first integer (ii) the 22nd integer (iii) the 75th integer and (iv) the last integer?



The first integer is stored at 0x004420c0. As each integer is 32 bits or 4 bytes, the address of the 2nd integer is 0x004420c0 + 4 = 0x004420c4, and so on. The nth integer is at address 0x004420c0 + 4(n-1). If n = 100, the 100th integer is at address 0x004420c0 +

4*99 = 0x004420c0 + 0x018c = 0x0044224C. The addresses of the 22nd (0x00442114) and 75th (0x004421E8) integers can be computed in a similar way.

Q5 Assuming that x is stored in R1, y in R2, z in R3 and the result in R0:

(i) Write ARM assembly language instructions to compute x + y + z.

```
ADD     R0, R1, R2      ; R0 = x + y
ADD     R0, R0, R3      ; R0 = x + y + z
```

(ii) Write ARM assembly language instructions to compute y – x - z.

(iii) Write ARM assembly language instructions to compute $x^2 + y^2 + z^2$.

(iv) Write ARM assembly language instructions to compute 5(x + y).

```
MOV     R0, #5  ; R0 = 5
ADD     R4, R1, R2      ; R4 = x + y
MUL     R0, R4, R0      ; R0 = 5*(x + y)
```

Need to work around the limitations of the MUL instruction - dst and src1 registers must not be the same and src2 cannot be an immediate value.

(v) Write ARM assembly language instructions to compute (x + y)(y – z).

(vi) Write ARM assembly language instructions to compute $3x^4 - 5x – 16y^4z^4$.

```
MOV     R0, #3  ; R0 = 3
MUL     R4, R1, R1      ; R1 = x^2
MUL     R0, R4, R0      ; R4 = 3x^2
MUL     R0, R4, R0      ; R0 = 3x^4
MOV     R4, #5  ; R4 = 5
MUL     R4, R1, R4      ; R4 = 5x
SUB     R0, R0, R4      ; R0 = 3x^4 - 5x
ADD     R4, R2, R2      ; R4 = 2y
MUL     R4, R3, R4      ; R4 = 2yz
MUL     R5, R4, R4      ; R5 = 4y^2z^2
MUL     R4, R5, R5      ; R5 = 16y^4z^4
SUB     R0, R0, R4      ; R0 = 3x^4 - 5x - 16y^4z^4
```

Need to work around the limitations of the MUL instruction - dst and src1 registers must not be the same and src2 cannot be an immediate value.