

# ECE 3140

Eric Berg

## Problem Set 1

Feb. 24, 2017

1. a)

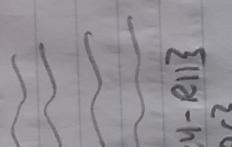
```
loop: CMP R2,#0
      BLT end
      LD RY [R0]
      LSL R5 RY // multiply x2
      LSL R6 R5 // multiply x2
      STR R6 [R1] // load output array
      SUB R2,R2,#1
      ADD R1,R1,#4 // skip to next array entry
      ADD R0,R0,#4 // skip to next array entry
      BL loop
```

end: WFI

b) 10 lines  $\rightarrow$  32 bits per instruction / 8 bits per byte  
 $(10 \cdot 32)/8 = 40$  bytes

c) BL multiply

```
multby4: PUSH {LR}
          PUSH {R4-R11}
          loop: ...
```



POP {R4-R11}  
POP {PC}

d) AND  $\underbrace{R_0}_{R_d}, \underbrace{PC}_{R_m}$ , # OxOF immediate

see A 7.7.8

This instruction is not legal because the manual states that the Rn register cannot be R13-R15, so because the  $R_n = PC = RS$  the operation is UNPREDICTABLE.

2. a) PC: R15  
LR: R14  
SP: R13  
arguments passed to function: R0-R3  
return value: R0

b) Stack after line 7

High Address	POPPED	POPPED	17
5	10	11 16 } caller saved	R1 2
2	128	11 R4 } callee saved	R2 3
128	1024	11 R5 } callee saved	R3 4
1024	48	11 R6 }	R4 3
48	3	-	R5 12
3	12	-	R6 2
12	2	1 mn 34	
2		1 dw 56	

- c) A double word argument can be passed to a function by passing a pointer to the first address where the first word is stored in memory

3. a) mistakes
- missing ";" after  $f = n - 1$  statement  $\rightarrow$  COMPILER ERROR
  - missing  $\{$  around 3 lines in this for loop  $\rightarrow$  NO COMPILER ERROR
  - Int f = n - 2; statement has no assignment but is added with later  $\Rightarrow$  SAP ERROR
  - missing  $\#include$  statement for function to be referenced  $\Rightarrow$  NO ERROR elsewhere

b) mylib.h

```
main.c
#include "mylib.h" // include "mylib.h"
int fib(int n) {
    if (n <= 1)
        return n;
    else
        return fib(n-1) + fib(n-2);
}

#endif
```

c) 102334155 hexadecimal [06197ECB]  
Cachework pad to make even      (local)  
Big Endian                            Little Endian

Small address	CB	CB7E1906
	7E	11
	19	[3414038790]
big address	06	value computed by

d) for (i = 0; i < 4; ++i) {  
 bytesTotal[i] = bytesTotal[bytes\_of\_b.length - i];  
 }

4. The bsearch function searches an array for a given value by splitting the array in half each time and searching within that subsection of the array. This search algorithm only works if the array is sorted from least to greatest.

```

    R1 = i
    R2 = j
    R3 = v
    R4 = h

bsearch: CMP R1, R2
        BEQ true

```

```

A00  R5, R1, R2
LSR R4, R5
LD R6 [R4]
CMP R6 R3
BEQ returnh
BGT bsearch1
BLE bsearch2

```

```

returnh: MOV R0, R4
        POP %LR
        BX LR
bsearch1: MOV R2, R4

```

```

true: LD R6 [R1]
      CMP R6 R3
      BEQ send1
      MOV R0, #-1
      POP %LR
      BX LR

```

```
send1: MOV R0, R1
```

```
POP %LR
```

```
BX LR
```

```
BL bsearch
```

```
bsearch2: ADD R4, R4, #-1
```

```
BL bsearch
```