

NAME: _____

SANTA CLARA UNIVERSITY
Department of Computer Engineering

COEN 020

Final Exam (Part 2)

Fall 2016

11. [5 pts ea] Convert each C function call into ARM assembly language.

C code	ARM Assembly
<pre>int8_t x8 ; int64_t y64 ; void f1(int8_t, int64_t) ; f1(x8, y64) ;</pre>	
<pre>uint64_t z64, f2(void) ; z64 = f2() ;</pre>	
<pre>int32_t count ; float real ; void f3(int32_t, float) ; f3(count, real) ;</pre>	
<pre>float result ; float f4(void) ; result = f4() ;</pre>	

NAME: _____

SANTA CLARA UNIVERSITY
Department of Computer Engineering

COEN 020

Final Exam (Part 2)

Fall 2016

12. [5 pts ea] Convert each C function definition into ARM assembly language.

C code	ARM Assembly
<pre>int64_t f5(int32_t s32) { return (int64_t) s32 ; }</pre>	
<pre>float f6(int8_t s8) { return (float) s8 ; }</pre>	
<pre>uint64_t f7(uint64_t u64) { return u64 + 1 ; }</pre>	
<pre>uint64_t f8(uint64_t u64) { return u64 << 4 ; }</pre>	

NAME: _____

SANTA CLARA UNIVERSITY
Department of Computer Engineering

COEN 020

Final Exam (Part 2)

Fall 2016

13. [5 pts ea] Convert each C function definition into ARM assembly language.

```
int32_t f9(int32_t x, int32_t a, int32_t b)
{
    if (x < a) return a ;
    if (x > b) return b ;
    return x ;
}
```

Do NOT use an IT block

```
int32_t f10(uint32_t score)
{
    if (score >= 60 && score <= 100)
        return 1 ;
    else
        return 0 ;
}
```

Do NOT use an IT block

NAME: _____

SANTA CLARA UNIVERSITY
Department of Computer Engineering

COEN 020

Final Exam (Part 2)

Fall 2016

14. [5 pts ea] Convert each C function definition into ARM assembly language.

int32_t f11(int64_t a64, int64_t b64) { return (a64 < b64) ? 1 : 0 ; }	Use an IT block
int32_t f12(int32_t a32) { int32_t f13(void) ; return f13() + a32 ; }	
int32_t f14(int32_t *p32, int32_k) { return *(p32 + k) ; }	

NAME: _____

SANTA CLARA UNIVERSITY
Department of Computer Engineering

COEN 020

Final Exam (Part 2)

Fall 2016

15. [5 Pts] Write a function in ARM assembly language that calculates the single-length (32-bit) product of two 32-bit unsigned operands. Unlike regular multiplication, the function should return the maximum 32-bit unsigned value (0xFFFFFFFF) if the single-length product overflows. The function prototype should be:

```
uint32_t Product(uint32_t a, uint32_t b) ;
```

16. [5 pts] Write a C inline function that returns the minimum of two integers. Use an extended asm statement inside the inline function so that the compiler is allowed to choose all of the registers. (Suggestion: Use an IT block for simplicity.)