NAME: _____

SANTA CLARA UNIVERSITY
Department of Computer Engineering

COEN 020 Midterm Exam Fall 2016

(Closed book & notes; No electronic devices)

Time Allowed: 1 hour

SCU's Academic Integrity Pledge

"I am committed to being a person of integrity. I pledge, as a member of the Santa Clara University community, to abide by and uphold the standards of academic integrity contained in the Student Conduct Code."

Signature:_____

Your Points:	
Max Points:	124
Your Score:	%

Points this page: _____ (16 max)

1. [1 pt ea] What is the range of each of the following data types?
    a. uint8_t: _____ to _____
    b. int8_t: _____ to _____

2. [3 pts ea] Convert the following <u>unsigned</u> integer representations:
    a. $4.5_6 \rightarrow ?_{10}$
    b. $27_{10} \rightarrow ?_5$
    c. $0.875_{10} \rightarrow ?_4$
    d. $101.101_2 \rightarrow ?_{16}$

3. [5 pts] Convert 1011.0110_2 from signed 2's complement to decimal.

4. [5 pts] Convert -50_{10} to an 8-bit 2's complement representation.

5. [4 pts total] Complete the binary addition, showing each of the carry and sum bits:

Carries:					0
A:		0	1	0	1
B:		0	0	1	1
A+B:					

6. [2 pts ea] Consider the following addition:

Carries:	**0**	**1**	?	?	?
A:		?	?	?	?
B:		?	?	?	?
A+B:		?	?	?	?

   a. Did an overflow occur if the operands were unsigned? _____

   b. Did an overflow occur if the operands were 2's complement? _____

Points this page: _____ (40 max)

7. [10 pts ea] Translate the following C into ARM assembly:

C Function	*ARM Assembly*

`uint64_t Promote(uint8_t x)`
`{`
`return (uint64_t) x ;`
`}`	

`int32_t One(int32_t x)`
`{`
`int32_t Two(void) ;`
 	
`return x + Two() ;`
`}`	

`int16_t Get(int16_t a[], int32_t k)`
`{`
`return a[k+1] ;`
`}`	

`int64_t *PlusK(int64_t *p, int32_t k)`
`{`
`return p + k ;`
`}`	

8. [10 pts ea] Translate the following C into ARM assembly:

C Function	*ARM Assembly*

`int32_t IsLT(int64_t a, int64_t b)`
`{`
`return (a < b) ? 1 : 0 ;`
`}`
	`// Use an IT block`

`int32_t Rem(int32_t a, int32_t b)`
`{`
`return a % b ;`
`}`	

`int32_t GetAndClear(int32_t *p)`
`{`
`int32_t temp ; // use a register`
 	
`temp = *p ;`
`*p = 0 ;`
`return temp ;`
`}`	

Points this page: _____ (20 max)

9. [10 pts ea] Translate the following C into ARM assembly:

C Function	*ARM Assembly*

`int32_t Valid(int32_t score)`
`  {`
`  if (0 <= score && score <= 100)`
`  return 1 ;`
`  else`
`  return 0 ;`
`  }`
	`// Do NOT use IT block`

`int32_t Note(uint32_t score)`
`  {`
`  if (score < 60 || score > 100)`
`  return 1 ;`
`  else`
`  return 0 ;`
`  }`
	`// Do NOT use IT block`