Version A
Solution

15F CST8216 Processor Architecture Term Test
Tuesday, November 17, 2015

Instructions:

1. Place your name and student number in the space provided.
2. Ensure that your copy of the test contains a total of 5 pages, including this cover page.
3. There are a total of 40 marks available.
4. Marks for each of the questions vary, and where noted you must show all of your work to be considered for full marks. Clearly indicate your answer to the questions.
5. A handout of the HCS12/9S12 Instruction Set Reference and other supplemental information is also provided with this test.
6. This is a closed book test.
7. No Scrap paper is permitted to be used during this term test – use the back of any page if you need extra space to solve problems.
8. You may take this test apart if you wish; I have a stapler at the front of the class.
9. If you have any questions, raise your hand.
10. Calculators are permitted to be used in this test.
11. Time limit: 50 minutes.

The use of electronic devices, with the sound turned on, during classes is strictly prohibited. In particular, cell phones are not to be used to communicate during a class. The use of any electronic devices during exams and mid-term tests, other than those sanctioned by the faculty in charge of the examination, is strictly prohibited.

Anyone caught using a prohibited device will be considered to have plagiarized, and will be treated as such in accordance with College Plagiarism Policy.

This test is worth 10% of your overall course mark.

Name _____

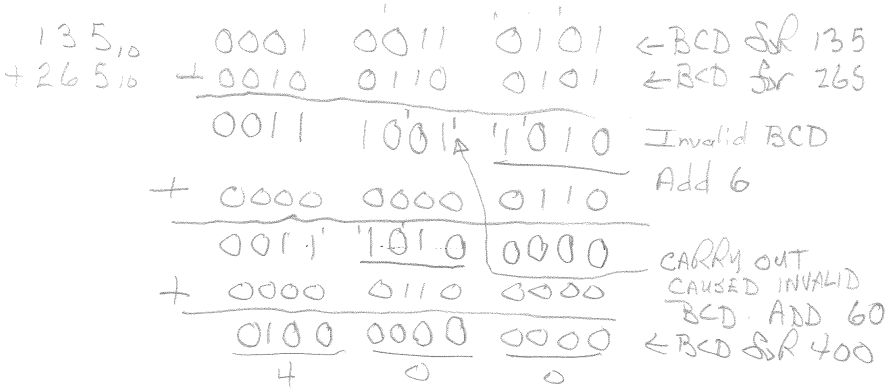
Student Number _____

Circle Your Lab Day/Time Tue 10 – 12 Wed 1 – 3 Wed 3 – 5 Fri 2 – 4

Number Conversions – (5 marks) Perform the following number conversions, placing your answers in the **Value** column:

Convert	To	Value
$F9 (unsigned)	Integer	249
$F9 (signed)	Integer	-7
$3FFF (unsigned)	Integer	16383
72	8-bit Hexadecimal (unsigned)	$48
-72	8-bit Hexadecimal (signed – use 2's complement)	$B8
-6	8-bit Hexadecimal (signed – use 2's complement)	$FA
128	Binary	%1000 0000
-128	Binary (use 2's complement)	%1000 0000
11111111 (signed)	Integer	-1
11111111 (unsigned)	Integer	255

BCD Arithmetic Methods – (5 marks) To confirm your understanding of BCD Arithmetic methods, represent 135_{10} and 265_{10} in BCD and then correctly perform BCD addition. **Show all steps** (including all annotations to the right of the additions) for full credit.



Program Tracing (10 marks)

Given the following program listing, trace the results (hexadecimal value of registers being asked) of each program step **after** the line of code has been executed, as per the example answer for SP. *(Instruction Set included with this Term Test).*

```
1  ; Trace1_VA.asm
2
3  ; Name          David Haley
4  ; S/N           Faculty
5  ; Date          7 Nov 2015
6
7  ; Purpose       To manually trace through given Assembly Language code
8
9  VALUE1  equ     $1234
10 VALUE2  equ     $21
11
12         org     $1000
13 Source  db      %11000011, $54, $82, 64
14 Source2 dw      $5678
15
16                                 ; All Values must be in Hexadecimal as per the
17                                 ; example for lds
18         org     $2000
19         lds     #$2000          ; S = $2000
20
21         ldd     #Value1         ; A = $12
22
23                                 ; B = $34
24
25         aba                     ; A = $46
26
27                                 ; B = $34
28
29         ldaa    Source+3        ; A = $40
30
31         ldab    $1000           ; B = $C3
32
33         ldx     Source2         ; X = $5678
34
35         ldx     #Source2        ; X = $1004
36
37         ldaa    #$01
38         ldab    #$02
39         exg     a,b             ; A = $02
40
41         ldy     Source+4        ; Y = $5678
42
43
44         end
45
```


Programming Concepts – Analysis of Code (10 marks)

Given the following source code listing, indicate the LEDs' output pattern for two complete iterations of the for (;;) [forever] loop by completing the table that follows the code listing. **Note that this is fully functional code, which correctly assembles and outputs a specific pattern on Port B's LEDs.**

To indicate that a specific LED is ON, place an X in the appropriate box in the table as per the following example (***which you should note is NOT part of the pattern produced by the code listing***). If a specific LED is OFF, leave the appropriate box in the table blank.

Time (msec)	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
3000	X			X			X	

```
;Walk_A_Bit_Structed_VA_Short_Version.asm
#include C:\68hcs12\registers.inc
; Program Constants
STACK   equ     $2000
; Delay_MS Constants
DVALUE  equ     #250    ; Delay value (base 10) 0 - 255 ms

        org     $1000   ; data area
LEDTABLE                ; values to display on LEDs
        db      $00, $18, $24, $42, $81
ENDTABLE

        org     $2000   ; program code
Start   lds     #STACK  ; stack location
        jsr     Config_SWs_and_LEDs

; Continually Flash LEDs with values from LEDTABLE, changing values every 250 ms
; for (;;)
Back    ldx     #LEDTABLE       ; point to first element of LEDTABLE
Again   ldaa    1,x+            ; get first value, increment pointer
        staa    portb           ; output value to LEDs
        ldaa    #DVALUE         ; delay value
        jsr     Delay_ms        ; delay routine
        cpx     #ENDTABLE       ; one less to do
        bne     Again           ; done yet?
        bra     Back            ; endless loop

; Predefined Subroutines Follow
#include C:\68HCS12\LIB\Config_SWs_and_LEDs.asm
#include C:\68HCS12\LIB\Delay_ms.asm
        end
```

Expanded view of LEDTABLE
$00, $18, $24, $42, $81

0000 0000
0001 1000
0100 0010
0010 0100
1000 0001

Indicate the LEDs' output pattern by completing this table (1 mark per correct line):

Time (msec)	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
0								
250				X	X			
500			X			X		
750		X					X	
1000	X							X
1250								
1500				X	X			
1750			X			X		
2000		X					X	
2250	X							X

$00
$18
$24
$42
$81
⋮

Program Tracing (10 marks)

Given the following program listing, trace the results (hexadecimal value of registers being asked) of each program step **after** the line of code has been executed, as per the example answer for SP. *(Instruction Set included with this Term Test).*

```
1  ; Trace2_VA.asm
2
3  ; Name          David Haley
4  ; S/N           Faculty
5  ; Date          8 Nov 2015
6
7  ; Purpose       To manually trace through given Assembly Language code
8          org     $1000
9  Data1   db       $32, $14, $98, $9D
10 Data2   db       $CE, $DA, $33, $7C, $01, $23, $02, $98
11
12                                 ; All Values must be in Hexadecimal as per the
13                                 ; example for lds
14         org     $2000
15         lds     #$2000          ; SP =?  $2000
16
17         ldy     #Data1          ; Y = ?  $1001
18
19         ldaa    0,y             ; A = ?  $32
20
21                                 ; Y = ?  $1000
22
23         ldab    1,y+            ; B = ?  $32
24
25                                 ; Y = ?  $1001
26
27         ldx     #Data2          ; X = ?  $1004
28
29         ldaa    4,x             ; A = ?  $01
30
31         inx
32
33         ldd     6,x+            ; A = ?  $DA
34
35                                 ; B = ?  $33
36
37                                 ; X = ?  $1008
38
39         swi
40         end
41
```


Version B

Solution

15F CST8216 Processor Architecture Term Test
Tuesday, November 17, 2015

Instructions:

1. Place your name and student number in the space provided.
2. Ensure that your copy of the test contains a total of 5 pages, including this cover page.
3. There are a total of 40 marks available.
4. Marks for each of the questions vary, and where noted you must show all of your work to be considered for full marks. Clearly indicate your answer to the questions.
5. A handout of the HCS12/9S12 Instruction Set Reference and other supplemental information is also provided with this test.
6. This is a closed book test.
7. No Scrap paper is permitted to be used during this term test – use the back of any page if you need extra space to solve problems.
8. You may take this test apart if you wish; I have a stapler at the front of the class.
9. If you have any questions, raise your hand.
10. Calculators are permitted to be used in this test.
11. Time limit: 50 minutes.

The use of electronic devices, with the sound turned on, during classes is strictly prohibited. In particular, cell phones are not to be used to communicate during a class. The use of any electronic devices during exams and mid-term tests, other than those sanctioned by the faculty in charge of the examination, is strictly prohibited.

Anyone caught using a prohibited device will be considered to have plagiarized, and will be treated as such in accordance with College Plagiarism Policy.

This test is worth 10% of your overall course mark.

Name _____

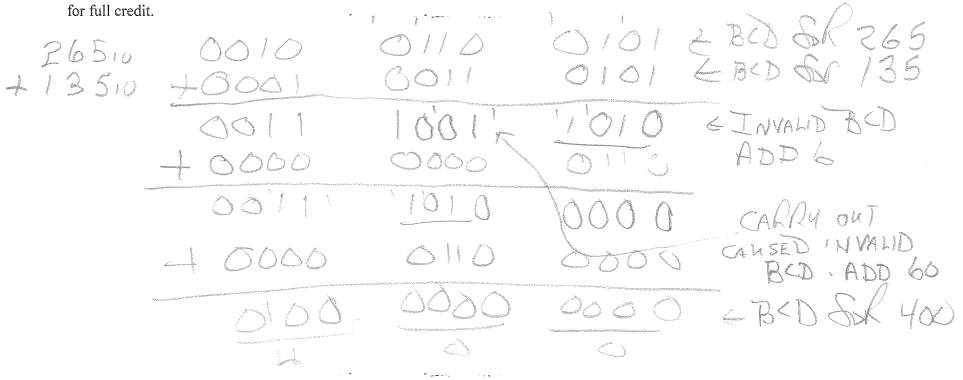
Student Number _____

Circle Your Lab Day/Time Tue 10 – 12 Wed 1 – 3 Wed 3 – 5 Fri 2 – 4

Number Conversions – (5 marks) Perform the following number conversions, placing your answers in the **Value** column:

Convert	To	Value
$F8 (unsigned)	Integer	248
$F8 (signed)	Integer	-8
$1FFF (unsigned)	Integer	8191
92	8-bit Hexadecimal (unsigned)	$5C
-92	8-bit Hexadecimal (signed – use 2's complement)	$A4
-6	8-bit Hexadecimal (signed – use 2's complement)	$FA
128	Binary	%10000000
-128	Binary (use 2's complement)	%10000000
11111110 (signed)	Integer	-2
11111110 (unsigned)	Integer	254

BCD Arithmetic Methods – (5 marks) To confirm your understanding of BCD Arithmetic methods, represent 265_{10} and 135_{10} in BCD and then correctly perform BCD addition. **Show all steps** (including all annotations to the right of the additions) for full credit.



Program Tracing (10 marks)

Given the following program listing, trace the results (hexadecimal value of registers being asked) of each program step **after** the line of code has been executed, as per the example answer for SP. *(Instruction Set included with this Term Test).*

```
1  ; Trace1_VB.asm
2  ; Name          David Haley
3  ; S/N           Faculty
4  ; Date          7 Nov 2015
5
6  ; Purpose       To manually trace through given Assembly Language code
7
8  VALUE1  equ     $5678
9  VALUE2  equ     $21
10
11         org     $1000
12 Source  db      %00111100, $82, $54, 128
13 Source2 dw      $1234
14
15                                 ; All Values must be in Hexadecimal as per the
16                                 ; example for lds
17         org     $2000
18         lds     #$2000          ; S = $2000
19
20         ldd     #Value1         ; A = $56
21
22                                 ; B = $78
23
24         aba                     ; A = $CE
25
26                                 ; B = $78
27
28         ldaa    Source+3        ; A = $80
29
30         ldab    $1000           ; B = $3C
31
32         ldx     Source2         ; X = $1234
33
34         ldx     #Source2        ; X = $1004
35
36         ldaa    #$02
37         ldab    #$01
38         exg     a,b             ; A = $01
39
40         ldy     Source+4        ; Y = $1234
41
42
43         end
```

Programming Concepts – Analysis of Code (10 marks)

Given the following source code listing, indicate the LEDs' output pattern for two complete iterations of the for (;;) [forever] loop by completing the table that follows the code listing. **Note that this is fully functional code, which correctly assembles and outputs a specific pattern on Port B's LEDs.**

To indicate that a specific LED is ON, place an X in the appropriate box in the table as per the following example (***which you should note is NOT part of the pattern produced by the code listing***). If a specific LED is OFF, leave the appropriate box in the table blank.

Time (msec)	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
3000	X			X			X	

```
;Walk_A_Bit_Structed_V2_Short_Version.asm
#include C:\68hcs12\registers.inc
; Program Constants
STACK   equ     $2000
; Delay_MS Constants
DVALUE  equ     #250    ; Delay value (base 10) 0 - 255 ms

        org     $1000   ; data area
LEDTABLE                ; values to display on LEDs
        db      $81, $42, $24, $18, $00
ENDTABLE

        org     $2000   ; program code
Start   lds     #STACK  ; stack location
        jsr     Config_SWs_and_LEDs

; Continually Flash LEDs with values from LEDTABLE, changing values every 250 ms
; for (;;)
Back    ldx     #LEDTABLE       ; point to first element of LEDTABLE
Again   ldaa    1,x+            ; get first value, increment pointer
        staa    portb           ; output value to LEDs
        ldaa    #DVALUE         ; delay value
        jsr     Delay_ms        ; delay routine
        cpx     #ENDTABLE       ; one less to do
        bne     Again           ; done yet?
        bra     Back            ; endless loop

; Predefined Subroutines Follow
#include C:\68HCS12\LIB\Config_SWs_and_LEDs.asm
#include C:\68HCS12\LIB\Delay_ms.asm
        end
```

Expanded view of LEDTABLE
$81, $42, $24, $18, $00

1000 0001 0010 0100 0010 0100 0001 1000

Indicate the LEDs' output pattern by completing this table (1 mark per correct line):

Time (msec)	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
0	X							X
250		X					X	
500			X			X		
750				X	X			
1000								
1250	X							X
1500		X					X	
1750			X			X		
2000				X	X			
2250								

$81 $42 $24 $18 $00

Program Tracing (10 marks)

Given the following program listing, trace the results (hexadecimal value of registers being asked) of each program step **after** the line of code has been executed, as per the example answer for SP. *(Instruction Set included with this Term Test).*

```
1  ; Trace2_VB.asm
2
3  ; Name          David Haley
4  ; S/N           Faculty
5  ; Date          8 Nov 2015
6
7  ; Purpose       To manually trace through given Assembly Language code
8          org     $1000
9  Data1   db       $02, $33, $FF, $06, $81
10 Data2   db       $04, $99, $9F, $40, $FE, $01, $23, $94, $01
11
12                                 ; All Values must be in Hexadecimal as per the
13                                 ; example for lds
14         org     $2000
15         lds     #$2000          ; SP =?  $2000
16
17         ldy     #Data1          ; Y = ?  $1000
18
19         ldaa    1,y+            ; A = ?  $02
20
21                                 ; Y = ?  $1001
22
23         ldab    0,y             ; B = ?  $33
24
25                                 ; Y = ?  $1001
26
27         ldx     #Data2          ; X = ?  $1005
28
29         ldaa    4,x             ; A = ?  $FE
30
31         inx
32
33         ldd     4,x+            ; A = ?  $99
34
35                                 ; B = ?  $9F
36
37                                 ; X = ?  $100A
38
39         swi
40         end
```