

ECEN 3213 Fall 2007 Exam 2

Name : \_\_\_\_\_ Solution \_\_\_\_\_ Date : \_8 Nov 2007

The questions in this exam refer to the 68HC11A8 single chip, unless otherwise noted in the question. Read the instructions carefully before completing the exam. All answers must be shown on these pages – no attached pages will be graded. This exam is open book, open notes, and you may use a calculator. You may not share books or calculators. You may not use a PDA, a cell phone, Tarot Cards, Ouija Board, or a laptop computer. Be careful to read the instructions, the questions, and the answers carefully. You should show your work because partial credit is possible.

For the following questions use the assembly program listing on the last two pages of this exam.

1. (5pts) For each of the addressing modes: give an example instruction, its address from the assembly listing, and the location of the operand:

Addressing mode	Example instruction	Address of example instruction	Location of operand for example instruction
Inherent	<u><b>inx</b></u> <u><b>nop</b></u> <u><b>nop</b></u> <u><b>lsra</b></u>	<u><b>\$E592</b></u> <u><b>\$E59A</b></u> <u><b>\$E58B</b></u> <u><b>\$E5CA</b></u>	<u><b>Register X</b></u> <u><b>No operand</b></u> <u><b>No operand</b></u> <u><b>Register A</b></u>
Immediate	<u><b>ldab #\$00</b></u> <u><b>ldaa #\$00</b></u> <u><b>ldx #\$0001</b></u>	<u><b>\$E59E</b></u> <u><b>\$E587</b></u> <u><b>\$E58C</b></u>	<u><b>Memory location \$E59F</b></u> <u><b>Memory location \$E588</b></u> <u><b>Memory locations \$E58D</b></u> <u><b>and \$E58E</b></u>
Direct	<u><b>ldaa \$02</b></u>	<u><b>\$E5AE</b></u>	<u><b>Memory location \$02</b></u>
Extended	<u><b>staa \$1003</b></u> <u><b>ldaa \$100A</b></u>	<u><b>\$E5DE</b></u> <u><b>\$E5E1</b></u>	<u><b>Register A</b></u> <u><b>memory location \$100A,</b></u> <u><b>which is Port E</b></u>
Indexed	<u><b>stab ,x</b></u> <u><b>stab 1,x</b></u>	<u><b>\$E58F</b></u> <u><b>\$E593</b></u>	<u><b>Register B</b></u> <u><b>Register B</b></u>

2. (3pts) How many bytes of EPROM are used to store the program instructions? **\$7A = 122**  
**\$E587 through \$E600**

⊖ 3. (3pts) How many bytes of the RAM are NOT used? 256-9 = 247  
**\$0000 through \$0008**

⊖ 4. (3pts) What is the latency in cycles? 173.  
[Note: partial credit for start and end addresses for first byte processed.]  
**\$E5CE first location and \$E5CD is end address.**

⊖ 5. (3pts) What is the throughput in Bytes/200cycles? ≈ 3.3 bytes/200 cycles.  
[Note: partial credit for start and end addresses of loop.]  
**\$E5A0 start of loop and \$E5B2 is end of loop**

6. (3 Points) Consider the program listing at the end of this exam. Which ports does it use?

PORT	Used for (input or output or bidirectional?)
<u>Port A \$1000</u>	<u>Input and Output(not bidirectional)</u> .
<u>Port B \$1004</u>	<u>Output</u> .
<u>Port C \$1003</u>	<u>Output</u> .
<u>Port E \$100A</u>	<u>Input</u> .

---

7. (2pts) What is the answer (show the bits for BCD representation)? 0100 0010

8. (3pts) What is the answer (in base 7)? 60

The following multiple-choice questions will be scored as 2 pts for each correct answer, -1 for an incorrect answer, and 0 for unanswered questions.

9. How many possible different values can be input on Port E of the 6811? **E**  
A) 8 B) 32 C) 64 D) 255 **E) 256** F) None of these

10. How many possible different values can be input on Port B of the 6811? **F**  
A) 8 B) 32 C) 64 D) 255 E) 256 **F) None of these**

11. If the stack is used only for storing subroutine return addresses, how deep can the subroutine nesting be for a 5 byte stack? **C**  
a. 0 b. 1 **c. 2** d. 3 e. 4 f. 8 g. None of these choices

12. For the same clock frequency, how much faster is a parallel 1 byte port than a 1 bit serial port with no start or stop bits? F  
 a. 0    b. 1    c. 2    d. 3    e. 4    **f. 8**    g. None of these choices
13. For the same clock frequency, how much faster is a parallel 1 byte port than a 1 bit serial port with 2 start bits and 1 stop bits? C  
 a. 0    b. 10    **c. 11**    d. 12    e. 4    f. 8    g. None of these choices
14. What does it mean when the PC register has the value 135? D  
 a. The next instruction to be fetched is in memory location 60135.  
 b. Something is probably wrong because 135 addresses RAM and instructions are usually in ROM.  
 c. This is a trick question because a 16-bit PC register cannot hold a value of less than 1024.  
**d. The next instruction to be fetched starts in memory location 135.**  
 e. None of the above.
15. Consider the instruction at address \$E589 at the end of this exam. How long does it take to fetch the instruction? B  
 A) 1 clock cycle  
**B) 2 clock cycles**  
 C) 3 clock cycles  
 D) None of the above
16. Consider the instruction at address \$E595 at the end of this exam. How long does it take to complete the instruction? D  
 A) 1 cycle  
 B) 2 cycles  
 C) 3 cycles  
**D) 4 cycles**  
 E) 5 cycles  
 F) None of the above
17. Which of the following calculations use the equation  $x = 2^n$ ? D  
 A) Number of words in address space for n bit address  
 B) Number of possible values for n bit binary number  
 C) Number of data bytes for n bit data word  
**D) A and B**  
 E) A and C  
 F) B and C  
 G) A, B, and C.

18. Consider the program listing at the end of this exam. How many bytes is the op code for the instruction at memory location \$E58C? **A**
- A) **1 byte**
  - B) 2 bytes
  - C) 3 bytes
  - D) None of the above
19. Consider the program listing at the end of this exam. How many bytes is the machine code for the instruction at memory location \$E58F? **B**
- A) 1 byte
  - B) **2 bytes**
  - C) 3 bytes
  - D) None of the above
- ☹ 20. Consider the program listing at the end of this exam. How many clock cycles does the 6811 take to fetch the instruction at memory location \$E5BE? **C**
- A) 1 clock cycle
  - B) 2 clock cycles
  - C) **3 clock cycles**
  - D) None of the above
21. Consider the program listing at the end of this exam. What is the address of the first instruction executed upon reset? **D**
- A) \$E587.
  - B) \$E589.
  - C) \$E5BE.
  - D) **\$E5CE**
  - E) \$B600
  - F) \$FFFE
  - G) \$FFFF.
  - H) None of the above.

The following True / False questions will be scored as 2 points for each correct answer, -1 for an incorrect answer and 0 for unanswered questions.

22.      **F** All digital computer systems have the CPU, some of the Memory, and the I/O ports on the same chip.
23.      **T** The program listed at the end of this exam uses I/O.
24.      **T** Timing diagrams can use four values, which are 1, 0, X, and Z..
- ☺25.      **T** A problem well stated is a problem half solved.
- ☺ 26.      **F** Hexadecimal numbers are more precise than binary numbers.
- ☺ 27.      **T** The shape for a decision operation in a flow chart is a diamond.
- ☺ 28.      **T** The shape for an I/O operation in a flow chart is a trapezoid.
29.      **F** The opcode for the bne instructions is \$26EC
30.      **T** The opcode for the jump to subroutine instruction is \$BD.
31.      **F** All digital computer systems have an analog I/O port.
32.      **F** The 6811 parallel input ports can only read data, not machine code instructions.
33.      **T** The 6811 has three general purpose registers.
34.      **T** Timing diagrams are used to show the relative order of events.
- ☺ 35.      **T** Direct Addressing can be used to address internal memory in a 68HC11A8.
36.      **T** Indexed addressing can be used to address any of the memory on the 6811.
- ☺ 37.      **T** This question number did not appear on Exam 2.
38.      **T** The stack can be in either on-chip RAM or off-chip RAM.
39.      **F** The 6811 analog I/O port is more accurate than the 6811 serial port.

⊗ 40. (13 pts) Consider the assembly program listing on the last two pages of this exam. Fill in the following table with the appropriate values after the given instruction has completed execution.

	\$E58B nop	\$E59A nop	\$E5D0 ldx # \$ABCD	\$E5D9 jmp start
[Reg A]	<u>\$00</u>	<u>\$00</u>	<u>XX</u>	<u>XX</u>
[Reg B]	<u>\$01</u>	<u>\$80</u>	<u>\$FF</u>	<u>\$FF</u>
[Reg X]	<u>\$ABCD</u>	<u>\$0008</u>	<u>\$ABCD</u>	<u>\$ABCD</u>
[Reg PC]	<u>\$E58C</u>	<u>\$E59B</u>	<u>\$E5D3</u>	<u>\$E587</u>
[Reg SP]	<u>\$0007</u>	<u>\$0007</u>	<u>XX</u>	<u>\$0007</u>
[Mem \$00]	<u>\$AB</u>	<u>\$AB</u>	<u>XX</u>	<u>\$AB</u>
[Mem \$01]	<u>\$CD</u>	<u>\$01</u>	<u>XX</u>	<u>\$CD</u>
[Mem \$02]	<u>XX</u>	<u>XX</u>	<u>XX</u>	<u>XX</u>
[Mem \$03]	<u>XX</u>	<u>\$02</u>	<u>XX</u>	<u>XX</u>
[Mem \$04]	<u>XX</u>	<u>\$04</u>	<u>XX</u>	<u>XX</u>
[Mem \$05]	<u>XX</u>	<u>\$08</u>	<u>XX</u>	<u>XX</u>
[Mem \$06]	<u>XX</u>	<u>\$10</u>	<u>XX</u>	<u>XX</u>
[Mem \$08]	<u>\$FF</u>	<u>\$40</u>	<u>XX</u>	<u>\$FF</u>

Copyright 2002-2003 Test EXecute And Simulate - Version 1.18 11/7/2007

\* ECEN3213 Exam 2

\* Input 8 bits with switches, 5 bits data, 3 bits control

\* Output 8 bits to be displayed on LCD .

\* Stop when pins 0-2 are all 0 on Input .

\* Output indicators to LEDs

```

$E587 8600      [ 2]( 0){pp      }start ldaa #$00
$E589 C601      [ 2]( 2){pp      }      ldab #$01
$E58B 01        [ 2]( 4){pf      }      nop
$E58C CE0001    [ 3]( 6){ppp     }      ldx #$0001
$E58F E700      [ 4]( 9){ppnw    }      stab ,x
$E591 58        [ 2](13){pf      }shift aslb
$E592 08        [ 3](15){pfn     }      inx
$E593 E701      [ 4](18){ppnw    }      stab 1,x
$E595 8C0008    [ 4](22){pppn    }      cpx  #$0008
$E598 2DF7      [ 3](26){pfn     }      blt shift
$E59A 01        [ 2](29){pf      }      nop
$E59B CEB600    [ 3](31){ppp     }restart ldx #$B600
$E59E C600      [ 2](34){pp      }      ldab #$00
$E5A0 A600      [ 4](36){ppnr    }step  ldaa ,x
$E5A2 B7100A    [ 4](40){pppw    }      staa $1000 *Output to Port A
$E5A5 BDE5BE    [ 6](44){ppprss  }      jsr reade
$E5A8 08        [ 3](50){pfn     }      inx
$E5A9 5C        [ 2](53){pf      }      incb
$E5AA C10A      [ 2](55){pf      }      cmpb #$0A
$E5AC 27ED      [ 3](57){pfn     }      beq restart
$E5AE 9602      [ 3](60){ppr     }      ldaa $02
$E5B0 8102      [ 2](63){pf      }      cmpa #$02
$E5B2 26EC      [ 3](65){pfn     }      bne step
$E5B4 BDE5DB    [ 6](68){ppprss  }      jsr wait4it
$E5B7 BDE5F1    [ 6](74){ppprss  }      jsr dumpN
$E5BA CF        [ 2](80){pf      }      stop
$E5BB 7EE5A0    [ 3](82){ppp     }      jmp step
* * * * *
$E5BE B61005    [ 4](85){pppr    }reade ldaa $100A *input from Port E
$E5C1 16        [ 2](89){pf      }      tab
$E5C2 D407      [ 3](91){ppr     }      andb $07
$E5C4 D702      [ 3](94){ppw     }      stab $02
$E5C6 94F8      [ 3](97){ppr     }      anda $F8
$E5C8 44        [ 2](100){pf     }      lsra
$E5C9 44        [ 2](102){pf     }      lsra
$E5CA 44        [ 2](104){pf     }      lsra
$E5CB 9701      [ 3](106){ppw    }      staa $01
$E5CD 39        [ 5](109){pfxuu  }      rts
* * * * *
$E5CE C6FF      [ 2](114){pp     }middle ldab #$FF
$E5D0 CEABCD    [ 3](116){ppp    }      ldx  #$ABCD
$E5D3 DF00      [ 4](119){ppww   }      stx  $00
$E5D5 8E0008    [ 3](123){ppp    }      lds  #$0008
$E5D8 37        [ 3](126){pfs    }      pshb
$E5D9 7EE587    [ 3](129){ppp    }      jmp  start

```

```

*****
$E5DC 860F          [ 2](132){pp          }wait4it ldaa #$0f
$E5DE B71003       [ 4](134){pppw        }          staa $1003 *Output to Port C
$E5E1 B6100A       [ 4](138){pppr        }          ldaa $100A *input from Port E
$E5E4 8407         [ 2](142){pp          }          anda #$07
$E5E6 27F4         [ 3](144){pfn         }          beq wait4it
$E5E8 9702         [ 3](147){ppw         }          staa $02
$E5EA D603         [ 3](150){ppr         }          ldab $03
$E5EC CA08         [ 2](153){pp          }          orab #$08
$E5EE F71000       [ 4](155){pppw        }          stab $1000 *Output to Port A
$E5F1 39           [ 5](159){pfxuu       }          rts
$E5F2 CEE398       [ 3](164){ppp         }dumpN ldx #$E398
$E5F5 D604         [ 3](167){ppr         }          ldab $4
$E5F7 A642         [ 4](170){ppnr        }next   ldaa $42,X
$E5F9 B71004       [ 4](174){pppw        }          staa $1004 *Output to Port B
$E5FC 08           [ 3](178){pfn         }          inx
$E5FD 5A           [ 2](181){pf          }          decb
$E5FE 26F7         [ 3](183){pfn         }          bne next
$E600 39           [ 5](186){pfxuu       }          rts

```

\*\*\*\*\*

```

$B600                                org $B600
$B600 0001030206070504               fcb $00,$01,$03,$02,$06,$07,$05,$04
$B608 0C0D0F0E0A                     fcb $0C,$0D,$0F,$0E,$0A
$B60D 2E                               fcc "."
$B60E 0000                             fcb $00,$00
$FFFE                                org $FFFE
$FFFE E5CE                             dc.w middle
$0000                                end

```

\*\*\*\*\*Symbol Table\*\*\*\*\*

```

dumpN          $E5F2
middle         $E5CE
next           $E5F7
reade          $E5BE
restart        $E59B
shift          $E591
start          $E587
step           $E5A0
wait4it        $E5DC

```

**Assembly successful**