Université d'Ottawa
Faculté de génie

École de science informatique
et de génie électrique

uOttawa
L'Université canadienne
Canada's university

University of Ottawa
Faculty of Engineering

School of Electrical Engineering
and Computer Science

Exercices pour préparation de l'examen final

Un extrait de problèmes du manuel du cours :
Fredrick M. Cady, Software and Hardware Engineering: Assembly and C programming for the Freescale HCS12 Microcontroller, Fredrick M. Cady., Oxford University Press, 2008.

7.1 Assume the memory display of the HCS12 shows 16 bytes starting at $0800

```
$0800:    B0 53 05 2B 36 89 00 FF FE 80 91 3E 77 AB 8F 7F
```

7.2 Use the contents of memory shown in problem 7.1 and give the results of the following instructions.

a.
```
ldx   $0800
```
X = ?

b.
```
ldy   $0802
```
Y = ?

c.
```
ldx   $0803
pshx
puld
```
X = ?, A = ?, B = ?

d.
```
ldd   $0800
ldx   $0802
exg   D,X
```
D = ?, X = ?

e. Assume X = $0800
```
ldd   $0A,X
```
D = ?

Solution

7.2 Use the contents of memory shown in problem 7.1 and give the results of the following instructions.

a.
```
ldx   $0800
```
X = ?

X = $B053

b.
```
ldy   $0802
```
Y = ?

Y = $052B

c.
```
ldx   $0803
pshx
puld
```
X = ?, A = ?, B = ?

X = $2B36, A = $2B, B = $36

d.
```
ldd   $0800
ldx   $0802
exg   D,X
```
D = ?, X = ?

D = $052B, X = $B053

e. Assume X = $0800
```
ldd   $0A,X
```
D = ?

D = $913E

7.4 Use the contents of memory shown in problem 7.1 and give the results of the following instructions.

a. Assume X = $0800
```
bset  0,X,$0F
```
($0800) = ?

b. Assume X = $0800
```
bset  6,X,$AA
```
EA = ?, (EA) = ?

c. Assume X = $0807
```
bclr  0,X,$AA
```
($0807) = ?

d. Assume Y = $0800
```
bclr  0,Y,$FF
```
($0800) = ?

Solution

7.4 Use the contents of memory shown in problem 7.1 and give the results of the following instructions.

a. Assume X = $0800

```
bset 0,X,$0F                ($0800) = ?
```

($0800) = $BF

b. Assume X = $0800

```
bset  6,X,$AA       EA = ?, (EA) = ?
```

EA = $0806, ($0806) = $AA

c. Assume X = $0807

```
bclr  0,X,$AA       ($0807) = ?
```

($0807) = $55

d. Assume Y = $0800

```
bclr  0,Y,$FF               ($0800) = ?
```

($0800) = $00

7.6 Use the contents of memory shown in problem 7.1 and give the results of the following instructions.

a.
```
ldaa  $0806
nega                A = ?, NZVC = ?
```

b.
```
ldaa  $0807
nega                A = ?, NZVC = ?
```

c.
```
neg   $0809         ($0809) = ?, NZVC = ?
```

d.
```
ldaa  $0806
coma                A = ?, NZVC = ?
```

e.
```
ldaa  $0807
coma                A = ?, NZVC = ?
```

f.
```
com   $0809         ($0809) = ?, NZVC = ?
```

Solution

7.6 Use the contents of memory shown in problem 7.1 and give the results of the following instructions.

a.
```
ldaa  $0806
nega
```
A = ?, NZVC = ?

A = $00, NZVC = 0100

b.
```
ldaa  $0807
nega
```
A = ?, NZVC = ?

A = $01, NZVC = 0001

c.
```
neg   $0809
```
($0809) = ?, NZVC = ?

($0809) = $80, NZVC = 1011

d.
```
ldaa  $0806
coma
```
A = ?, NZVC = ?

A = $FF, NZVC = 1001

e.
```
ldaa  $0807
coma
```
A = ?, NZVC = ?

A = $00, NZVC = 0101

f.
```
com   $0809
```
($0809) = ?, NZVC = ?

($0809) = $7F, NZVC = 0001

7.14 The memory display shows:

```
4000:  08  29  3F  7F   86  99  A0  64 ...
```

and the current value in the X register is $4000

Give the results of the following instructions: [a, k]

a.
```
ldaa  $4000
```
A = ?

b.
```
ldab  $4,x
```
B = ?

c.
```
ldy   $4001
```
Y = ?

Now, with the value in the B register as computed in part b, the following instructions are executed:

```
cmpb  #$50
bgt   Someplace
```

d. Is the branch taken?

Solution

7.14 The memory display shows:

```
4000:  08  29  3F  7F  86  99  A0  64 ...
```

and the current value in the X register is $4000

Give the results of the following instructions: [a, k]

a. `ldaa $4000` A = ?
A = $08

b. `ldab $4,x` B = ?
B = $86

c. `ldy $4001` Y = ?
Y = $293F

Now, with the value in the B register as computed in part b, the following instructions are executed:

```
cmpb  #$50
bgt   Someplace
```

d. Is the branch taken?
No, because B = $86 (-122$_{10}$) is not greater than $50 (+80$_{10}$)

7.20 For each of the following instructions, assume A = $C9 and the NZVC bits are 1001. Give the result in A and the NZVC bits for each of the following instructions. [k]

a. `lsla` b. `lsra` c. `asla` d. `asra` e. `rola` f. `rora`

a. `lsla` *A =* *NZVC =*
b. `lsra` *A =* *NZVC =*
c. `asla` *A =* *NZVC =*
d. `asra` *A =* *NZVC =*
e. `rola` *A =* *NZVC =*
f. `roar` *A =* *NZVC =*

Solution

7.20 For each of the following instructions, assume A = $C9 and the NZVC bits are 1001. Give the result in A and the NZVC bits for each of the following instructions. [k]

a. `lsla` b. `lsra` c. `asla` d. `asra` e. `rola` f. `rora`

a. `lsla` *A = $92, NZVC = 1001*
b. `lsra` *A = $64, NZVC = 0011*
c. `asla` *A = $92, NZVC = 1001*
d. `asra` *A = $E4, NZVC = 1001*
e. `rola` *A = $93, NZVC = 1001*
f. `roar` *A = $E4, NZVC = 1001*

7.44 Assume Data1 and Data2 in the table that follows are +100$_{10}$ and -100$_{10}$ respectively. For each of the instructions give the results in the A, B, Y and D accumulators. [a, b]

Instruction	A	B	Y	D
`ldaa Data1`				
`sex a,y`				
`ldaa Data2`				
`tfr a,d`				
`emuls`				
`std Data3`				

Solution

7.44 Assume Data1 and Data2 in the table that follows are $+100_{10}$ and -100_{10} respectively. For each of the instructions give the results in the A, B, Y and D accumulators. [a, b]

Instruction	A	B	Y	D
`ldaa Data1`				
`sex a,y`				
`ldaa Data2`				
`tfr a,d`				
`emuls`				
`std Data3`				

Instruction	*A*	*B*	*Y*	*D*
`ldaa Data1`	*$64*	*xx*	*$xxxx*	*$64xx*
`sex a,y`	*$64*	*xx*	*$0064*	*$64xx*
`ldaa Data2`	*$9C*	*xx*	*$0064*	*$9Cxx*
`tfr a,d`	*$FF*	*$9C*	*$0064*	*$FF9C*
`emuls`	*$D8*	*$F0*	*$FFFF*	*$D8F0*
`std Data3`	*$D8*	*$F0*	*$FFFF*	*$D8F0*

8.2 For each of the logic statements, give the appropriate HCS12 code to set the condition code register and to branch to the ELSE part of an IF-THEN-ELSE. Assume P and Q are 8-bit, signed numbers in memory locations P and Q. [c, k]

a. IF P >= Q

b. IF Q > P

c. IF P = Q

Solution

8.2 For each of the logic statements, give the appropriate HCS12 code to set the condition code register and to branch to the ELSE part of an IF-THEN-ELSE. Assume P and Q are 8-bit, signed numbers in memory locations P and Q. [c, k]

a. IF P >= Q

```
        ; IF P >= Q
              ldaa  P
              cmpa  Q
              blt   ELSE_PART
```

b. IF Q > P

```
        ; IF Q > P
              ldaa  Q
              cmpa  P
              ble   ELSE_PART
```

c. IF P = Q

```
        ; IF P = Q
              ldaa  P
              cmpa  Q
              bne   ELSE_PART
```

8.6 Insert code to implement the following structured design immediately after each design comment. Assume the following structured design is just a small segment of an overall program.

Assume the following 8-bit unsigned variable data allocations have been made and have been initialized in some other part of the program. [c, k]

```
Temp3:    DS.B  1
Temp4:    DS.B  1
Temp5:    DS.B  1

; Implement the following design
; WHILE Temp3 > Temp4
; DO
;  Temp4 = Temp4 + 1
;  Temp5 = 2 * Temp5
; ENDWHILEDO
```

Solution

8.6 Insert code to implement the following structured design immediately after each design comment. Assume the following structured design is just a small segment of an overall program.

Assume the following 8-bit unsigned variable data allocations have been made and have been initialized in some other part of the program. [c, k]

```
Temp3:    DS.B  1
Temp4:    DS.B  1
Temp5:    DS.B  1

; Implement the following design
; WHILE Temp3 > Temp4
; DO
;  Temp4 = Temp4 + 1
;  Temp5 = 2 * Temp5
; ENDWHILEDO
enddo:
```

```
      ; Implement the following design
      ; WHILE Temp3 > Temp4
      while_start:
            ldaa  Temp3
            cmpa  Temp4
            bls   enddo
      ; DO
      ;  Temp4 = Temp4 + 1
            inc   Temp4
      ;  Temp5 = 2 * Temp5
            als   Temp5
            bra   while_start
      ; ENDWHILEDO
      enddo:
```

8.10 Write HCS12 assembly language code for the following pseudocode design assuming K1, K2, and K3 are 8-bit, signed or unsigned, numbers in memory locations K1, K2, and K3. Assume memory has been allocated for these data. [c, k]

```
; WHILE K1 does not equal $0d
while_start:
;  DO
;     IF K2 = K3
;     THEN
;        K1 = K1 + 1
;        K2 = K2 - 1
;     ELSE
;        K1 = K1 - 1
;     ENDIF K2 = K3
;  ENDO
; ENDWHILEDO
```

Solution

8.10 Write HCS12 assembly language code for the following pseudocode design assuming K1, K2, and K3 are 8-bit, signed or unsigned, numbers in memory locations K1, K2, and K3. Assume memory has been allocated for these data. [c, k]

```
; WHILE K1 does not equal $0d
while_start:
;  DO
;     IF K2 = K3
;     THEN
;        K1 = K1 + 1
;        K2 = K2 - 1
;     ELSE
;        K1 = K1 - 1
;     ENDIF K2 = K3
;  ENDO
; ENDWHILEDO
```

```
     ; WHILE K1 does not equal $0d
     while_start:
                ldaa   K1
                cmpa   #$0d
                beq    endo
     ;    DO
     ;       IF K2 = K3
                ldaa   K2
                cmpa   K3
                bne    elsepart
     ;       THEN
     ;          K1 = K1 + 1
                inc    K1
     ;          K2 = K2 - 1
                dec    K2
                bra    endif
     elsepart:
     ;       ELSE
     ;          K1 = K1 - 1
                dec    K1
     endif:
     ;       ENDIF K2 = K3
                bra    while_start
     ;    ENDO
     endo:
     ; ENDWHILEDO
```

10.6 Write an assembly language module to which the C program passes three integer values and receives the sum of these as an int in return. [c, k]

int function(int arg1, int arg2, int arg3);

```
     ;***********************************************
     ; Assembly C callable function
     ;   int function( int arg1, int arg2, int arg3 );
     ;
             XDEF  function
     ;***********************************************
     ; Assembly language interface
     ;
```

Solution

10.6 Write an assembly language module to which the C program passes three integer values and receives the sum of these as an int in return. [c, k]

int function(int arg1, int arg2, int arg3);

```
;*************************************************
; Assembly C callable function
;   int function( int arg1, int arg2, int arg3 );
;
        XDEF  function
;*************************************************
; Assembly language interface
;
function:
; arg1 and arg2 are passed on the stack and arg3 in D
; Add all int arguments and return the result
        addd   4,SP   ; Add arg1 to arg3
        addd   2,SP   ; Add arg2
; D now has the sum of the three inputs and is the
; return value
;
; Return to the calling program
        rts
;*************************************************
```

11.2. Give the data register addresses for ports A, B, AD, T, P, S, and M.

Solution

11.2. Give the data register addresses for ports A, B, AD, T, P, S, and M. [a]

PTA = $0000, PTB = $0001, PTAD = $0270, PTT = $0240, PTP = $0258, PTS = $0248, PTM = 0250

12.4. What advantage does a vectored interrupt system have over a polled interrupt system?

Solution

12.4. What advantage does a vectored interrupt system have over a polled interrupt system? [a]

The time taken to start the interrupt service routine is shorter in a vectored system because extra code does not have to be executed to poll all interrupting devices.

12.8. In the HCS12, why does the CPU automatically push all registers on the stack before transferring control to the user's interrupt service routine?

Solution

12.8. In the HCS12, why does the CPU automatically push all registers on the stack before transferring control to the user's interrupt service routine? [a]

To preserve the machine state or context.

12.10. Which instruction is used to globally mask interrupts?

Solution

12.10. Which instruction is used to globally mask interrupts? [k]

SEI

12.12. Interrupts are masked when you get to the interrupt service routine - true or false?

12.14. How are interrupts unmasked if the CLI instruction is not executed in the interrupt service routine?

12.16. What address does the HCS12 use to find the address of an interrupt service routine for a timer overflow?

Solution

12.12. Interrupts are masked when you get to the interrupt service routine - true or false? [a]

true

12.13. In the HCS12 interrupt service routine, you MUST unmask interrupts with the CLI instruction before returning - true or false? [a]

false

12.14. How are interrupts unmasked if the CLI instruction is not executed in the interrupt service routine? [g]

The RTI instruction pulls all registers, including the CCR, from the stack when returning to the interrupted program. The I bit will be cleared at that time because it was clear when it was pushed.

12.15. In the HCS12, what is the difference between an interrupt mask bit and an interrupt enable bit? [a, g]

The interrupt mask acts globally on all interrupting sources; a 1 masks (stops) interrupts and 0 unmasks (allows) interrupts.
An interrupt enable bit acts locally on an individual interrupting source; a 1 enables the interrupt and 0 disables.

12.16. What address does the HCS12 use to find the address of an interrupt service routine for a timer overflow? [k]

$FFDE:FFDF

14.2. What is wrong with the following code to get the 16-bit value of the TCNT register?

```
ldab $45     ; Get the low byte
ldaa $44     ; Get the high byte
```

14.4. How is the TCNT clock prescaler programmed?

Solution

14.2. What is wrong with the following code to get the 16-bit value of the TCNT register? [b, k]

```
ldab $45     ; Get the low byte
ldaa $44     ; Get the high byte
```

If the TCNT register is incrementing at each bus clock cycle, by the time you get through reading the low byte, it may have overflowed, causing the high byte to be incremented by one. This is less likely to cause an error than reading the high byte first as in 14.1 but it still could happen.

14.3. How should you read the 16-bit TCNT value? [k]

LDD $44 latches the 16-bit data from the TCNT register when the value is stable.

14.4. How is the TCNT clock prescaler programmed? [a]

Bits PR2, PR1 and PR0 in the TSCR2 register may be set by the programmer.

14.6. When is the timer overflow flag set?

14.8. What timing resolution can be achieved with the output compare?

Solution

14.6. When is the timer overflow flag set? [k]

When the TCNT register rolls over from $FFFF to $0000.

14.7. How is the timer overflow flag reset? [k]

By the software writing a one to bit-7 of TFLG2 register.

14.8. What timing resolution can be achieved with the output compare? [a]

*(1 bus clock period) * (the prescaler value in TSCR2).*

14.9. The TCNT register is receiving an 8 MHz clock. [a, c, k]

a. How many clock cycles will constitute a delay of 5.8 ms?

*5.8 ms * $8x10^6$ clocks/sec = 46,400*

b. If you are using an output compare with interrupts to delay 5.8 ms, can this be done without multiple interrupts?

Yes, the required delay is less than 8.192 ms assuming an 8 MHz clock.

14.10. Give the name of the bit, the name of the register that it is in, the register's address, which bit, and the default or reset state of the bit for each of the following: [k]

a. What bit indicates that a comparison has been made on Output Compare 2?

C2F, in the TFLG1 register at $004E, bit 2. The reset state is 0.

b. What bit enables the Output Compare 2 interrupt?

C2I, in the TIE register at $004C, bit 2. The reset state is interrupts disabled.

c. What bits are used to set the Output Compare 3 I/O pin high on a successful comparison?

OM3 and OL3 in the TCTL2 register at $0049, bits 7 and 6. The reset state is for the timer to be disconnected from the output pin.

Solution

14.10. Give the name of the bit, the name of the register that it is in, the register's address, which bit, and the default or reset state of the bit for each of the following: [k]

a. What bit indicates that a comparison has been made on Output Compare 2?

C2F, in the TFLG1 register at $004E, bit 2. The reset state is 0.

b. What bit enables the Output Compare 2 interrupt?

C2I, in the TIE register at $004C, bit 2. The reset state is interrupts disabled.

c. What bits are used to set the Output Compare 3 I/O pin high on a successful comparison?

OM3 and OL3 in the TCTL2 register at $0049, bits 7 and 6. The reset state is for the timer to be disconnected from the output pin.

14.11. Write a small section of code to set the Output Compare 2 I/O pin to toggle on every comparison. [c]

```
ldaa  #%00010000 ; Set OM2, OL2 to 01
staa  $49        ; Write to TCTL2 register
```

14.12. Write a small section of code to enable Output Compare 7 to set bits PT7, PT6 and PT5 to one on the next successful comparison. [c]

```
ldaa  #%11100000  ; Set bits 7,6,5 high
staa  $43         ; Write the Output Compare data register OC7D
staa  $42         ; Enable bits 7, 6, and 5 in OC7M mask
register
```

14.14. What two registers control which data bits are output when the Output Compare 7 flag is set?

14.16. Write a short section of code demonstrating how to reset the Input Capture 2 Flag C2F.

14.18. What bits in what registers must be set to enable the Pulse Accumulator Input Edge interrupt.

Solution

14.14. What two registers control which data bits are output when the Output Compare 7 flag is set? [c]

The Output Compare 7 Data (OC7D) register located at $43 and the Output Compare 7 Mask (OC7M) register located at $42.

14.15. How does the programmer select the active edge for Input Capture 2? [k]

Timer Control Register 4 (TCTL4) at $4B contains bits to select the edge used for the input capture. Bits 5 and 4 control the edge for IC2 by the following truth table:

EDG2B	*EDG2A*	*Edge*
0	*0*	*Capture Disabled*
0	*1*	*Capture on rising edges only*
1	*0*	*Capture on falling edges only*
1	*1*	*Capture on any edge, rising or falling*

14.16. Write a short section of code demonstrating how to reset the Input Capture 2 Flag C2F. [c, k]

```
ldaa  #%00000100  ; C2F position
staa  $4E         ; Write to TFLG1
```

14.18. What bits in what registers must be set to enable the Pulse Accumulator Input Edge interrupt.

PAI, bit-0 in PACTL, $60 must be set.

14.20. Give the name of the bit, the name of the register that it is in, the register's address, which bit, and the default or reset state of the bit for each of the following:

a. What bit enables PWM channel 0?

b. What bit must be set if you want an active-high pulse for the duty cycle time on PWM channel 0?

c. What bit must be set to combine PWM channels 0 and 1 to create a 16-bit PWM channel?

d. What bit must be set to switch the PWM output from channel 3 to port T, bit-3?

Solution

14.20. Give the name of the bit, the name of the register that it is in, the register's address, which bit, and the default or reset state of the bit for each of the following: [k]

a. What bit enables PWM channel 0?

PWME0, in the PWME register at $00E0, bit-0. The reset state is 0 (disabled).

b. What bit must be set if you want an active-high pulse for the duty cycle time on PWM channel 0?

PPOL0, in the PWMPOL register at $00E1, bit-0. The reset state is 0.

c. What bit must be set to combine PWM channels 0 and 1 to create a 16-bit PWM channel?

CON01, in the PWMCTL register at $00E5, bit-4. The reset state is 0.

d. What bit must be set to switch the PWM output from channel 3 to port T, bit-3?

MODRR3, in the MODRR register at $0247. The reset state is 0.

Intermediate:

14.21. Write a short section of code demonstrating how to enable the real-time interrupt and to set the nominal rate to 16.384 ms assuming an 8 MHz oscillator clock. [c, k]

```
bset    $3B,#%01100111 ; Set RTR6:RTR0 in RTICTL
bset    $37,#%10000000 ; Write to RTIFLG in CRGFLG to reset
                       ; real-time interrupt flag
bset    $38,#%10000000 ; Write to RTIE in CRGINT to enable
                       ; real-time interrupts
```

14.26. The exact total divisor (627.44) is not achievable in problem 14.25. With the total divisor = 628, PWMPER0 = 255 and PWMDTY0 = 64, what is the actual period and duty high time?

Solution

14.24. The infamous 68FC12 processor has a 12-bit timer subsystem similar to the HCS12's. It has a 12-bit TCNT register with one 12-bit output compare register and similar flags and controls. Assume that it has a 1 MHz clock. [c, k]

a. What is the interval between timer overflows?

$2^{12} = 4096 * 10^{-6} = 4.096$ ms

b. Assuming the present value of the TCNT register is $D18 what value should be loaded into the output compare register to create a delay of

(i) 100 microseconds?

100 microseconds = 100 counts = $64 counts; $D18 + $64 = $D7C

(ii) 1 millisecond?

1 ms = 1000 counts = $3E8 counts; $D18 + $3E8 = $100 (modulo 4096 addition)

15.2 An SCI is transmitting data at the following Baud rates. The format is eight data bits, no parity, one stop bit. For each case, what is the maximum number of characters per second that can be transmitted?

a. 56 kBaud

b. 9600 Baud

Solution

15.2 An SCI is transmitting data at the following Baud rates. The format is eight data bits, no parity, one stop bit. For each case, what is the maximum number of characters per second that can be transmitted? [a]

a. 56 kBaud

*56,000 bits/sec * 1 character/10 bits = 5600 characters/sec.*

b. 9600 Baud

*9600 bits/sec * 1 character/10 bits = 960 characters/sec.*

15.4 For the SCI, give the name of the bit, the name of the register it is in, the register's address, which bit, and the default or reset state of the bit for each of the following:

a. What bit enables an interrupt when the transmit buffer is empty?

b. What bit enables an interrupt when the transmitter has completely emptied its serial shift register?

c. What bit enables interrupts by the SCI receiver?

Solution

15.4 For the SCI, give the name of the bit, the name of the register it is in, the register's address, which bit, and the default or reset state of the bit for each of the following: [k]

a. What bit enables an interrupt when the transmit buffer is empty?

TIE, in the SCICR2 register at $00CB, bit-7; TDRE interrupts are disabled.

b. What bit enables an interrupt when the transmitter has completely emptied its serial shift register?

TCIE, in the SCICR2 at $00CB, bit-6; transmit complete interrupts are disabled.

c. What bit enables interrupts by the SCI receiver?

RIE, in the SCICR2 at $00CB, bit-5; receiver interrupts are disabled.

15.6 What different status information do the SCI status bits TDRE and TC give?

Solution

15.6 What different status information do the SCI status bits TDRE and TC give? [k]

TDRE indicates that the last character has been transferred from the SCIDRL transmit register to the output shift register. When TDRE is set you may write another byte to the SCIDRL register. TC, transmission complete, indicates that the last character has been shifted out of the shift register. When TC is set all data have been transmitted from the SCI. This occurs 10 bit-times after TDRE is set.

15.10 On which port and which bits are the serial communications interface (SCI) transmitted and received data?

Solution

15.10 On which port and which bits are the serial communications interface (SCI) transmitted and received data? [a]

RxD = port S, bit-0, TxD = port S, bit-1

17.2 What is a sample-and-hold?

Solution

17.2 What is a sample-and-hold? [a]

A device, usually a capacitor and an electronic switch, with a very short aperture time that samples, and then holds constant, the analog signal for the A/D.

17.6 An 8-bit successive approximation, general purpose (not HCS12) A/D converter has a 1 MHz clock. What is the maximum frequency it can convert without aliasing?

17.8 An A/D converter is required to digitize a -5 to +5 volt analog signal to a resolution of 10 mV. How many bits are required?

Solution

17.6 An 8-bit successive approximation, general purpose (not HCS12) A/D converter has a 1 MHz clock. What is the maximum frequency it can convert without aliasing? [a]

*1 sample/8 clocks * 10^6 clocks/sec = 125 kilo samples/sec*
*fsample = 2 * fmax, therefore fmax = 62.5 kHz.*

17.7 An A/D converter is required to digitize a 5 KHz sinusoidal waveform. What is the maximum allowable conversion time for the A/D? (Assume a sample-and-hold circuit is being used.) [a, c]

0.1 ms

17.8 An A/D converter is required to digitize a -5 to +5 volt analog signal to a resolution of 10 mV. How many bits are required? [a, c]

$2^n \geq 10\ V/10\ mV = 1000.\ n = 10$ *bits*

17.9 The A/D is programmed to convert a sequence of four channels in 8-bit continuous conversion mode. What is the maximum frequency signal on PAD0 that can be converted without aliasing (ignore aperture time effects, assume the final sample time is two ATD clocks and the ATD clock is 2 MHz). [b, k]

The conversion time for any one channel is 12 (2 + 2 + 8) ATD clock periods giving the tital sequence conversion time of 48 clocks = 24 µsec. Thus the sampling frequency for any one channel is 41.67 kHz and the Nyquist frequency is 20.83 kHz.

17.10 The analog input ranges from 1 volt to 4 volts.

a. What should V_{RH} and V_{RL} be?

b. What is the resolution for an 8-bit conversion?

c. The analog result register shows $56. What is the analog voltage?

Solution

17.10 The analog input ranges from 1 volt to 4 volts. [a, k]

a. What should V_{RH} and V_{RL} be?

$V_{RH} = 4\ V$ and $V_{RL} = 1\ V$.

b. What is the resolution for an 8-bit conversion?

11.7 mV

c. The analog result register shows $56. What is the analog voltage?

*$\$56 = 86_{10} * 11.7\ mV + 1.00\ v = 2.00\ v$*

17.11 The analog input is 0 - 5 volts and $V_{RH} = 5$ V, $V_{RL} = 0$ V. The A/D reading is $24. What is the analog input voltage for a 10-bit conversion? [b, k]

*Resolution = 5 V/1024 = 4.88 mV. $\$24 = 36_{10}$. Input voltage is 36 * 4.88 mV = 0.175 V*

17.12 A 10-bit successive approximation A/D converter has the following specs:

Minimum conversion time: 10^{-4}; input voltage: -5 to +5 volts [a, c, k]

a. What is the maximum frequency that can be sampled without aliasing?

b. For this frequency, what is the aperture time required so that errors in sampling are less than plus or minus ½ least significant bit?

c. What is the resolution of this A/D in volts?

Solution

17.12 A 10-bit successive approximation A/D converter has the following specs:

Minimum conversion time: 10^{-4}; input voltage: -5 to +5 volts [a, c, k]

a. What is the maximum frequency that can be sampled without aliasing?

$$f_{MAX} = \frac{1}{2 * 10^{-4}} = 5000 Hz$$

b. For this frequency, what is the aperture time required so that errors in sampling are less than plus or minus ½ least significant bit?

$$t_{AP} = \frac{1}{2\Pi * f_{MAX} * 2^n} = 31.08 ns$$

c. What is the resolution of this A/D in volts?

$$Resolution = \frac{V_{MAX}}{2^n} = \frac{10}{1024} = 9.8 mV$$

17.13 What sample rate must be used to sample a signal with a maximum frequency of 4 kHz [b, k]

$f_{sample} = 2 * f_{max} = 8$ kHz.

17.14 For problem 17.13, assuming an 8-bit A/D converter, what is the maximum aperture time allowed? For a 10-bit converter? [b,k]

$t_{AP} = 155$ ns for 8-bits and 38 ns for 10-bits.