DIGITAL- OCH MIKRODATORTEKNIK , U2

10-04-07

09.00 – 13.00

Tillåtna hjälpmedel: Instruktionslista PIC16F877A
Lista på registeruppsättningen i PIC16F877A
Datablad TTL-kretsar 74-serien

Fullständiga lösningar skall inlämnas. Konstruktionerna skall ritas om inget annat anges. Minimalitet krävs, om inget annat anges.

Skriv namn på alla inlämnade blad.

Skrivningsformuläret skall inlämnas.

Maxpoäng:	80
Digitalteknik	40
Mikrodatorteknik	40

DU MÅSTE KLARA MINST 12 POÄNG PÅ VARJE DEL FÖR ATT BLI GODKÄND PÅ TENTAMEN.

TOTALT FÖR HELA TENTAMEN GÄLLER:

För godkänt (betyg 3) krävs	32 p
För betyg 4 krävs	48 p
För betyg 5 krävs	64 p

Frågor under tentamen: Börje Dellstrand tel. 167122
0702-986358

Lycka till!

Bilaga: INTCON-registret
STATUS-registret
Instruction Set

Uppgift D1 **10 poäng**

D1.1 Översätt det binära heltalet, Z, till basen 10.
(2p)

$Z = 10101100_2$

D1.2 Översätt följande tal, Y, som är representerat i tvåkomplementsform, till
(2p) basen 10.

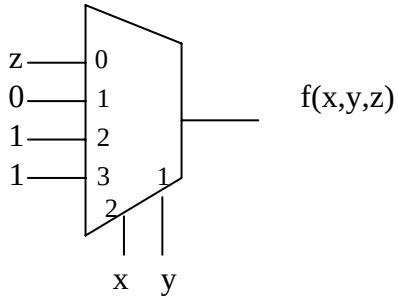
$Y = 101011_{2C}$

D1.3 SP-formen för en boolesk funktion kan skrivas som $f(x, y, z, w) = \Sigma(1,3,9,11)$
(2p) x är den mest signifikanta variabeln. Funktionen kan också skrivas som:

a) $x' \cdot w$
b) $y \cdot w$
c) $y' \cdot z'$
d) $y' \cdot w$

D1.4 Insignalerna till multiplexern är x, y och z.
(2p) Vilken logisk funktion realiserar kopplingen ?

a) $z + x' \cdot y$
b) $x' \cdot z' + z \cdot y$
c) $x \cdot y' + z'$
d) $x \cdot y' + z'$
e) $x + y' \cdot z$

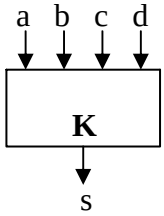


D1.5 I dessa uppgifter förutsätts 8 bitars ordlängd och 2-komplementering.
(2p) Ange svaret på följande additioner, dels i binär form och dels som decimaltal.
Ange också om overflow inträffar eller ej. Motivera väl.

a) 7E + 2A
b) FA + F9

Uppgift D2 **10 poäng**

s = 1 om i mönstret <a b c d> finns minst två nollor intill varandra.

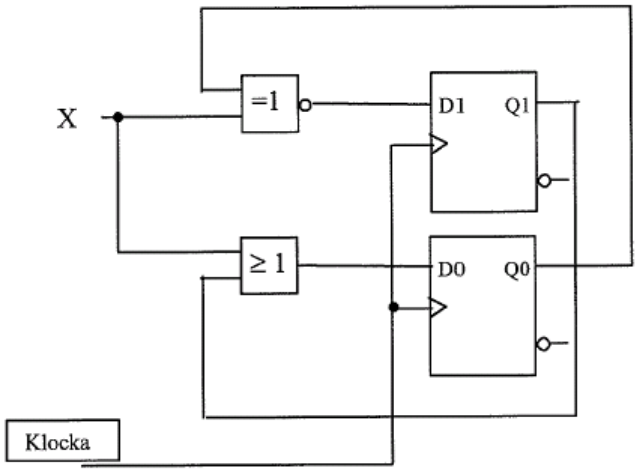


a b c d	s
0 0 0 0	1
0 0 0 1	1
0 0 1 0	1
0 0 1 1	1
0 1 0 0	1
0 1 0 1	0
0 1 1 0	0
0 1 1 1	0
1 0 0 0	1
1 0 0 1	1
1 0 1 0	0
1 0 1 1	0
1 1 0 0	1
1 1 0 1	0
1 1 1 0	0
1 1 1 1	0

Realisera med
a) NAND-grindar och ev. inv.
b) en 8/1-MUX och ev. en inv.
c) två 3/8-AVK, 74LS138, och en valfri grind.
Rita alla anslutningar.
Insignalernas inverser antas tillgängliga.

Uppgift D3 **5 poäng**

Rita en tillståndsgraf för nedanstående sekvenskrets.



Uppgift D4 **10 poäng**

Konstruera en synkron sekvenskrets av typ Moore som svarar mot följande beskrivning: Kretsen skall ha **en** ingång x och **en** utgång u. Insignalen består av en serie ettor och nollor i godtycklig ordning. Utsignalen u skall vara 1 om de senaste tre mottagna värdena på insignalen varit ettor. I övriga fall skall den vara 0.

Ex. x 0 1 1 0 0 1 0 1 1 1 0 0 1 1 1 0 1 0 . . .
u 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 . . .

Använd JK-vippor och valfria grindar.

Uppgift D5 10 poäng

a) Räknaren 74LS163 skall kopplas som en modulo 9-räknare, så att den räknar upp från värdet 3 till värdet 11 (decimalt), och därefter börjar om. Räkning startas/stoppas med en insignal R. Rita en figur, som visar kopplingen (alla insignaler inkopplade).
Valfria grindar får användas.

b) Konstruera en modulo 28-räknare (28 olika tillstånd) m h a två stycken räknare 74LS163 plus valfri logik. Räkning startas/stoppas med en insignal R. En utsignal u ska visa 1 under 1CP efter 28 klockpulser. u = 0 f.ö.
Valfri tillståndskod får användas. Rita komplett schema.

Uppgift M6 5 poäng

Kombinera en term i den vänstra kolumnen med tillhörande fras(er) i den högra kolumnen.

1. Assemblator	A. asynkron seriell överföring
2. RISC	B. synkroniseringsmetod
3. Refresh	C. datariktningsregister
4. Baud	D. översättningsprogram
5. Random accessminne	E. alfanumerisk kod
6. Nibble	F. förenklad instruktionsuppsättning
7. PROM	G. alltid samma åtkomsttid
8. Paritetsbit	H. krävs för dynamiska minnen
9. ASCII	I. enhet för signaleringshastighet
10. TRISB	J. generella register
	K. 7-bitars tabell
	L. kan ej raderas
	M. här sker beräkningar
	N. 4 bitar

Uppgift M7 4 poäng

En microcontroller måste kunna kommunicera med yttre enheter.

Programmerad IN/UT-kommunikation (**polling**) och avbrottsstyrd IN/UT-kommunikation (**interrupt**) är två vanliga sätt som används.

Förklara de två sätten samt ange för- och nackdelar med respektive metod.
Skillnaden mellan metoderna ska klart framgå.

Uppgift M8 6 poäng

Ett minne, av typ EEPROM, är organiserat som 4k*8

a) Hur många adresspinnar har minnet?
b) Hur många (exakt, i decimal form) ord rymmer minnet?
c) Hur stor är ordlängden ?
d) Hur många bitar rymmer minnet?
e) Beskriv med några meningar vad ett EEPROM är. Vad står förkortningen EEPROM för?

Uppgift M9 5 poäng

a) Beskriv vad stacken är och vad den används till.
b) Vad är ALU:ns uppgift i CPU:n ?
c) Skriv en instruktion som ettställer Z-flaggan.
d) Beskriv bitsekvensen som måste sändas för att överföra ASCII-tecknet 'C' (43_{16}) seriellt genom att använda udda paritet, 7 databitar, 1 stoppbit, 9600 baud. Rita figur

Uppgift M10 5 poäng

Ange värdena i filregistren I och J **efter** exekvering av vidstående program.
Innehållet i arbetsregistret w skall också anges.

```
                LIST        P=16F877A
                INCLUDE     <P16F877A.INC>

I               EQU         0X20
J               EQU         0X21

                ORG         0X00
                MOVLW       3
                MOVWF       I
                MOVLW       5
                MOVWF       J
                INCF        I,F
                MOVLW       2
                ADDWF       I,W
                MOVWF       J
                MOVF        I,W
                ADDLW       2
                MOVWF       J
                MOVF        I,W
                SUBWF       J,W
                BTFSC       STATUS,Z
                INCF        I,F
STOP            GOTO        STOP

                END
```

Uppgift M11 5 poäng

Räkna antalet nollor i 8 bits variabeln INDATA (adress 0x20) och spara resultatet i variabeln UTDATA (adress 0x21).
Ex INDATA = 1111 1001 ska ge UTDATA = 2.
Skriv ett väl kommenterat program i PIC-assembler .

Uppgift M12 **5 poäng**

Antag att vi har ett kretskort där PORTC är ansluten till 8 insignaler (knappar där en nedtryckt knapp ger en etta) och PORTB är ansluten till 8 lysdioder så att de tänds när man ettställer motsvarande bit i PORTB.

```
PICSUB    MOVWF    TEMP
          MOVF     PORTC,W
          ANDLW    70
          XORLW    70
          BTFSC    STATUS,Z
          GOTO     ON
          CLRW
          GOTO     SLUT
ON        MOVLW    20
SLUT      MOVWF    PORTB
          MOVF     TEMP,W
          RETURN
```

Antag att ovanstående program körs om och om igen.
Här följer fem påstående om vad som händer då. Ange för varje påstående, vilket som är sant och vilket som är falskt !

1. Om alla knappar trycks ner, så kommer en lysdiod att tändas.
2. Om man trycker ner flera än tre knappar, så kommer alltid en lysdiod att tändas.
3. Man kan aldrig få mer än en lysdiod att lysa.
4. Det finns inget sätt att tända fler än två lysdioder.
5. Om man inte trycker på någon knapp, så lyser ändå en lysdiod.

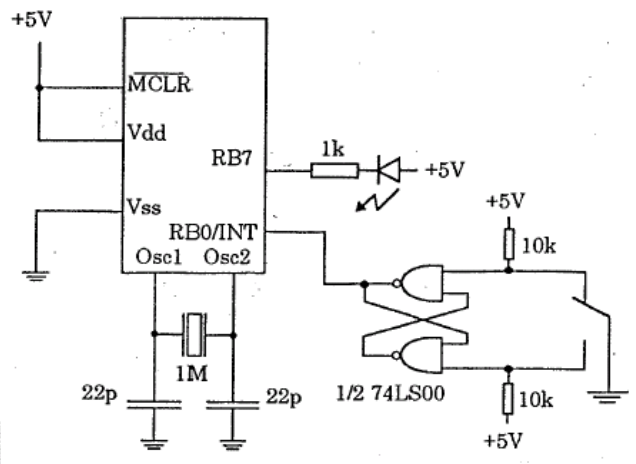
Uppgift M13 **5 poäng**

```
          LIST     P=16F874
          INCLUDE <P16F874.INC>

          ORG      0X00
          GOTO     MAIN

          ORG      0X04
          BCF      INTCON,1
          COMF     PORTB,1
          RETFIE

MAIN      BSF      STATUS,RP0
          MOVLW    7F
          MOVWF    TRISB
          BCF      STATUS,RP0
          BSF      INTCON,7
          BSF      INTCON,4
LOOP      GOTO     LOOP
          END
```



a) Beskriv ingående (inte bara vad varje instruktion utför) vad nedanstående program utför. Se bilaga.

b) Vad kallas denna metod när det gäller kommunikation med yttre enheter ?

REGISTER 2-3: INTCON REGISTER (ADDRESS 0Bh, 8Bh, 10Bh, 18Bh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
bit 7							bit 0

bit 7 **GIE:** Global Interrupt Enable bit
1 = Enables all unmasked interrupts
0 = Disables all interrupts

bit 6 **PEIE:** Peripheral Interrupt Enable bit
1 = Enables all unmasked peripheral interrupts
0 = Disables all peripheral interrupts

bit 5 **T0IE:** TMR0 Overflow Interrupt Enable bit
1 = Enables the TMR0 interrupt
0 = Disables the TMR0 interrupt

bit 4 **INTE:** RB0/INT External Interrupt Enable bit
1 = Enables the RB0/INT external interrupt
0 = Disables the RB0/INT external interrupt

bit 3 **RBIE:** RB Port Change Interrupt Enable bit
1 = Enables the RB port change interrupt
0 = Disables the RB port change interrupt

bit 2 **T0IF:** TMR0 Overflow Interrupt Flag bit
1 = TMR0 register has overflowed (must be cleared in software)
0 = TMR0 register did not overflow

bit 1 **INTF:** RB0/INT External Interrupt Flag bit
1 = The RB0/INT external interrupt occurred (must be cleared in software)
0 = The RB0/INT external interrupt did not occur

bit 0 **RBIF:** RB Port Change Interrupt Flag bit
1 = At least one of the RB7:RB4 pins changed state; a mismatch condition will continue to set the bit. Reading PORTB will end the mismatch condition and allow the bit to be cleared (must be cleared in software).
0 = None of the RB7:RB4 pins have changed state

Status registret (adress 03h, 83h)

bit7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
IRP	RP1	RP0	TO'	PD'	Z	DC	C

Bit7: IRP:Register Bank Select bit (används för indirekt adressering)
0=Bank 0, 1 (00h-FFh)
1=Bank 2, 3 (100h-1FFh)
IRP biten används inte på PIC16F8X och bör vara satt till noll för at garantera framåtkompatibilitet med större PIC'ar.

Bit5-6 RP1:RP0: Register Bank Select bit (andvänds för direkt adressering.
00=Bank 0 (00)-7Fh)
01=Bank 1 (80h-FFh)
10=Bank 2 (100h-17Fh)
11=Bank 3 (180h-1FFh)
Varje bank innehåller 128 bytes. Endast RP0 används på PIC16F87. RP1 bör vara satt till noll för att garantera framåtkompatibilitet med större PIC'ar.

Bit4: TO': Time out bit
1=Efter påslag av kretsen, efter en CLRWDT instruktion eller SLEP instruktion.
0=En Watch Dog Timer time-out har inträffat.

Bit3: PD': Power-down bit
1=Efter en power-up eller då instruktionen CLWDT har utförts.
0=Då en SLEEP instruktion har utförts

Bit2: Z: Zero bit
1=Resultatet av en räkneoperation eller en logisk operation har blivit noll.
0=Resultatet av en räkneoperation eller en logisk operation inte harblivit noll.

Bit1: DC: Digit carry/borrow' bit (för ADDWF och ADDLW instruktioer)(För borrow' är det tvärt om)
1=En carry bit från fjärde biten från msb har inträffat.
0=ingen carry bit från fjärde biten från msb har inträffat.

Bit0: C: Carry/borrow bit (för ADDWF och ADDLW instruktioner)
1=En carry bit har ramlat ut från msb av resultatbyten
0=Ingen carry bit har ramlat ut från msb av resultatbyten
OBS när det gäller borrow(lånebit) är förhållandena omkastade

TABLE 13-2: PIC16F87X INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-Bit Opcode MSb			LSb	Status Affected	Notes
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C,DC,Z	1,2
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	1,2
CLRF	f	Clear f	1	00	0001	lfff	ffff	Z	2
CLRW	-	Clear W	1	00	0001	0xxx	xxxx	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	1,2
DECF	f, d	Decrement f	1	00	0011	dfff	ffff	Z	1,2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1,2,3
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	1,2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1,2,3
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	1,2
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	1,2
MOVWF	f	Move W to f	1	00	0000	lfff	ffff		
NOP	-	No Operation	1	00	0000	0xx0	0000		
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	1,2
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	1,2
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C,DC,Z	1,2
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff		1,2
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		1,2
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		3
LITERAL AND CONTROL OPERATIONS									
ADDLW	k	Add literal and W	1	11	111x	kkkk	kkkk	C,DC,Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
CALL	k	Call subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDT	-	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}$,$\overline{PD}$	
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk		
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLW	k	Move literal to W	1	11	00xx	kkkk	kkkk		
RETFIE	-	Return from interrupt	2	00	0000	0000	1001		
RETLW	k	Return with literal in W	2	11	01xx	kkkk	kkkk		
RETURN	-	Return from Subroutine	2	00	0000	0000	1000		
SLEEP	-	Go into standby mode	1	00	0000	0110	0011	$\overline{TO}$,$\overline{PD}$	
SUBLW	k	Subtract W from literal	1	11	110x	kkkk	kkkk	C,DC,Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	