

## **DIGITAL- OCH MIKRODATORTEKNIK , U2**

**07-12-15**

**09.00 – 13.00**

Tillåtna hjälpmedel: Instruktionslista PIC16F874  
Lista på registeruppsättningen i PIC16F874  
Schema på PIC 16F874  
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. 0702-986358

Lycka till!

Bilaga: STATUS-registret, INTCON-registret

**Uppgift D1 (10p)**

- a) Addera binärt talen 35 och -24. Anta två-komplementform och 8-bitars ordlängd. Svara i hexadecimal form. Visa tillvägagångssätt.
- b) Antag att det är 400 personer i en klass. Varje person ska få ett unikt bitmönster. Hur många bitar går det minst åt?
- c) Skriv talen 7 och -7 i 4-bitars 2-komplementform
- d) Vilket är största positiva tal som kan representeras med 8-bitar 2-komplementform?
- e) Vilket är det mest negativa tal som kan representeras med 8-bitar 2-komplementform?
- f) Sök minimala uttrycket för

$$f(x_3 x_2 x_1 x_0) = \sum(3, 5, 7, 11) + d(6, 15)$$

och ange också uttrycket på NAND-form.

**Uppgift D2 (10p)**

Konstruera ett kombinatoriskt nät, som realiserar vidstående tabells Booleska funktion.

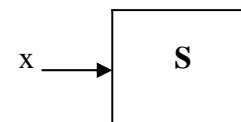
Använd

- a) NAND-grindar. Inverterade signaler antas tillgängliga.
- b) en 8/1 MUX 74LS151 och ev. en inverterare.  
Bonuspoäng med 4/1 MUX + ev. en inv.
- c) en 3/8-avkodare 74LS138 och en valfri grind.  
Visa den fullständiga inkopplingen av avkodaren.

D	C	B	A	f
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

**Uppgift D3 (10p)**

Konstruera ett sekvensnät S så att för  $x=0$  räknar det 00, 01, 10, 00, . . . (modulo 3-räknare) och för  $x=1$  räknar det 00, 01, 10, 11, 00, . . . (modulo 4-räknare). Använd JK-vippor och NAND-grindar.



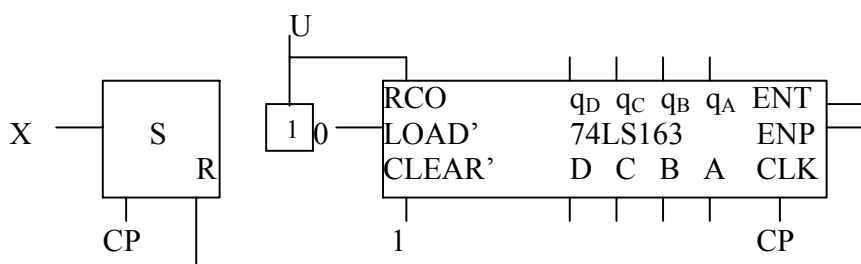
**Uppgift D4 (10p)**

Ett synkront sekvensnät med en ingång X och en utgång U, enl. fig. nedan har följande beteende: Var 11:e gång en ”skur” av ett jämnt antal ettor förekommit i insekvensen, skall en etta genereras på utgången U.

Ex.: I sekvensen 101100011110010110 förekommer en sådan sekvens tre gånger.

Utsignalen U ska vara 1 under ett klockpulsintervall.

Ange lämpligt startvärde för räknaren. Konstruera nätet S med ingången X och utgången R (=räknevillkor). Använd D-vippor och valfria grindar.



**Uppgift M5 (5p)**

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

- |                       |                                      |
|-----------------------|--------------------------------------|
| 1. Assembler          | A. asynkron seriell överföring       |
| 2. RISC               | B. synkroniseringsmetod              |
| 3. Refresh            | C. datarikttningsregister            |
| 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 M6 (5p)**

Vad är skillnaden mellan ett assemblerdirektiv och en instruktion? Ge exempel på en av varje. Ange också hur mycket minne det går åt för exemplet på assemblerdirektiv resp. exemplet på en instruktion.

**Uppgift M7 (2p)**

Nämnen  
a) fördel med SRAM jämfört med DRAM  
b) nackdel med SRAM jämfört med DRAM

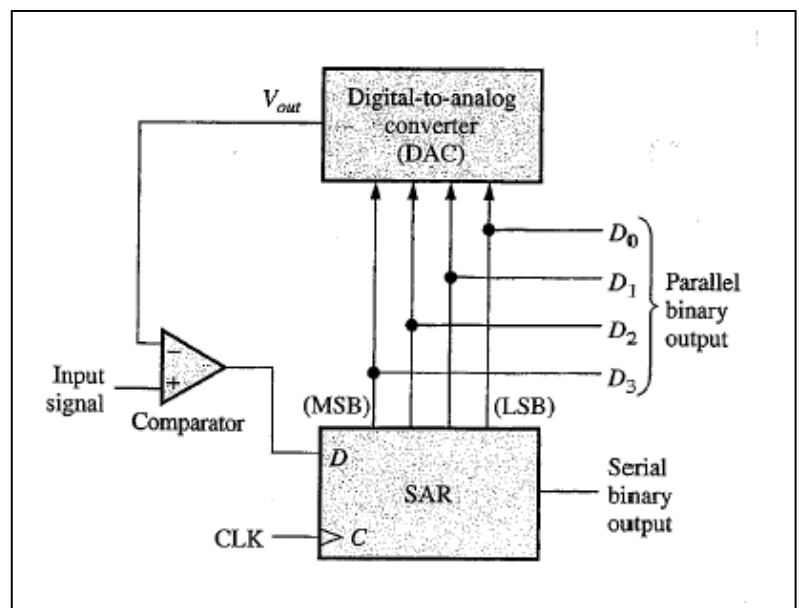
**Uppgift M8 (4p)**

Ett minne har kapaciteten 64 kbyte och alltså ett minnesområde 0000-FFFF<sub>sexton</sub>. Ange adressområdet i hexadecimal form för

- a) de första 4 kbyte i minnet
- b) de sista 8 kbyte i minnet
- c) Minnet ska byggas upp med minneskapslar 32 kbit (8kx4).  
Hur många minneskapslar kommer att behövas? Motivera!

**Uppgift M9 (4p)**

Figuren visar en A/D-omvandlare. Ange namn, fördelar och nackdelar för denna typ av A/D-omvandlare. Beskriv också, utgående från fig. hur omvandling sker.



### Uppgift M10 (4p)

Betrakta följande programkod:

```
ORG 0X00
GOTO START

ORG 0X10
START MOVLW 0X17
CALL ASUB
STOP GOTO STOP

ASUB XORLW 0XFF
RETURN
```

Förklara vad programmet utför och ange vad W-registret innehåller efter det att programmet exekverats.

### Uppgift M11 (6p)

a) Initiera PortB efter följande hårdvaruritning. PortB är kopplad till följande enheter:

Knapp0	Knapp1	LED0	LED1	Knapp2	LED2	LED3	LED4
RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0

b) Skriv kod för att läsa av knapp2. Om knapp2 är nedtryckt (aktiv låg) så ska LED2 och LED4 tändas, utan att påverka resten av registret. Om knapp2 ej är nedtryckt ska avläsningen av knapp2 upprepas.

### Uppgift M12 (5p)

Skriv två program för att kontrollera statusen på en ingång IN7 ansluten till RB7 pinnen och visar IN7:s status på en LED ansluten till RB1 pinnen (PortB). De två programmen skall implementeras så att:

- IN7 kontrolleras med polling
- IN7 kontrolleras med avbrott (interrupt). Bilaga: INTCON-registret

### Uppgift M13 (5p)

Skriv ett program som hittar det största talet (binär representation) i TABELL. Placera det största talet på adressen MAX.

TABELL	ADDWF	PCL,1	;Positionsnummer
	RETLW	0X12	;0
	RETLW	0X32	;1
	RETLW	0X33	;2
	RETLW	0X22	;3
	RETLW	0X10	;4
	RETLW	0X63	;5
	RETLW	0X42	;6
	RETLW	0X65	;7
	RETLW	0X34	;8
	RETLW	0X11	;9

**Bilaga**

**REGISTER 2-1: STATUS REGISTER (ADDRESS 03h, 83h, 103h, 183h)**

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
					bit 0		

- bit 7 **IRP:** Register Bank Select bit (used for indirect addressing)  
 1 = Bank 2, 3 (100h-1FFh)  
 0 = Bank 0, 1 (00h-FFh)
  - bit 6-5 **RP1:RP0:** Register Bank Select bits (used for direct addressing)  
 11 = Bank 3 (180h-1FFh)  
 10 = Bank 2 (100h-17Fh)  
 01 = Bank 1 (80h-FFh)  
 00 = Bank 0 (00h-7Fh)  
 Each bank is 128 bytes.
  - bit 4  **$\overline{TO}$ :** Time-out bit  
 1 = After power-up, CLRWD $\overline{T}$  instruction or SLEEP instruction  
 0 = A WDT time-out occurred
  - bit 3  **$\overline{PD}$ :** Power-down bit  
 1 = After power-up or by the CLRWD $\overline{T}$  instruction  
 0 = By execution of the SLEEP instruction
  - bit 2 **Z:** Zero bit  
 1 = The result of an arithmetic or logic operation is zero  
 0 = The result of an arithmetic or logic operation is not zero
  - bit 1 **DC:** Digit carry/borrow bit (ADDWF, ADDLW, SUBLW, SUBWF instructions)  
 (for borrow, the polarity is reversed)  
 1 = A carry-out from the 4th low order bit of the result occurred  
 0 = No carry-out from the 4th low order bit of the result
  - bit 0 **C:** Carry/borrow bit (ADDWF, ADDLW, SUBLW, SUBWF instructions)  
 1 = A carry-out from the Most Significant bit of the result occurred  
 0 = No carry-out from the Most Significant bit of the result occurred
- Note:** For borrow, the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate (RRF, RLF) instructions, this bit is loaded with either the high, or low order bit of the source register.

**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	TMR0IE	INTE	RBIE	TMR0IF	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 **TMR0IE:** 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 **TMR0IF:** 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