

Uppgift 1 (5p)

- Skriv som ett hexadecimalt tal 90_{10} . 0.5p
- Skriv som ett binärt tal $C5A_{16}$. 0.5p
- Skriv talet -73_{10} som ett binärt tal på 2-komplementform med 8 bitars 1p representation.
- Realisera funktionen $f = a'b + b'c$ med NOT, AND och OR. 1p
Inga inverterade signaler finns tillgängliga.
- Realisera funktionen OR, dvs. $f = a + b$ med endast 2-ingångars NAND. 1p
Inga inverterade signaler finns tillgängliga.
- Förenkla uttrycket $a + bc + bc' + a'b + a'b'$ så långt det är möjligt. 1p

Uppgift 2 (6p)

Funktionen U är given.

Konstruera U med:

- ett minimalt NAND-grindnät
(insignalernas inverser antas tillgängliga)
- två st 3/8 AVK (74LS138) plus en valfri grind
(fullständig inkoppling visas)
Bonus 1p om uppgiften löses med **en** 3/8 AVK plus en valfri grind
- en 8/1 MUX plus ev. en inverterare

x_3	x_2	x_1	x_0	U
0	0	0	0	0
0	0	0	1	0
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	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

Uppgift 3 (5p)

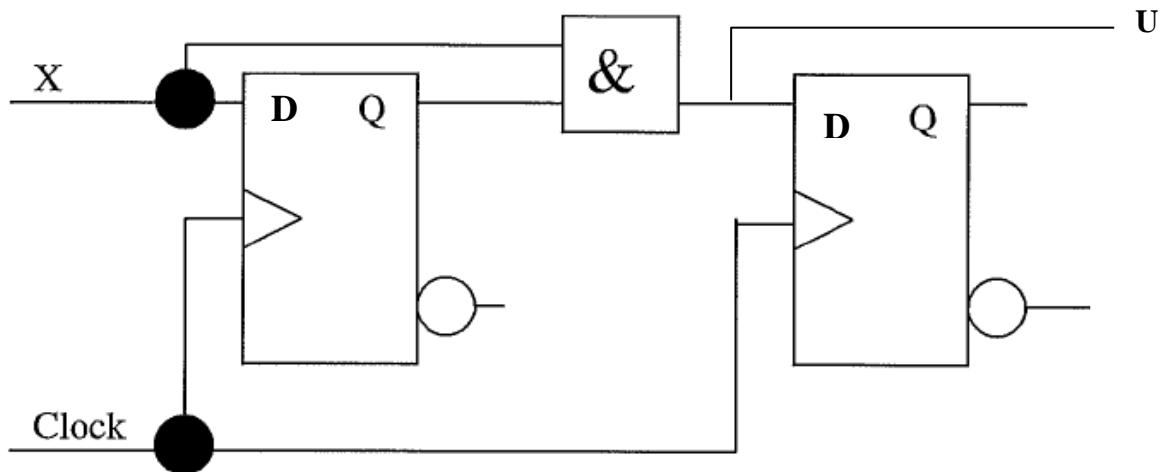
Konstruera en tillståndsmaskin som ger utsignal $u = 1$ för varje följd 11 hos insignalen x .
Obs! Överlappning accepteras **ej!**

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

Vid realiseringen används D-vippor samt NAND-grindar.

Uppgift 4 (4p)

Analysera den klockade synkrona sekvenskretsen med D-vippor enligt schemat nedan, genom att rita tillståndsgraf.



Uppgift 5 (4p)

Konstruera med hjälp av räknare av typ 74LS163 och valfria grindar

- en modulo 120-räknare (120 tillstånd). Valfri tillståndskod får användas. Efter 120 steg ska utsignalen $U_T = 1$ under 1 CP.
- en modulo 6-räknare (6 tillstånd) med räknesekvensen 1, 2, 3, 4, 5, 6, 1, 2, ...

Räknarna ska vara försedda med räknevillkor, insignalen R.
Då $R=0$ ska räknaren stå stilla och då $R=1$ ska framstegning ske på varje klockpuls.
Räknarna ska vara konsekvent synkroniserade.
Rita konstruktionerna.

Uppgift 6 (6p)

6.1 Vad blir resultatet då instruktionen ANDLW 0x34 har utförts? Arbetsregistret w innehåller från början det binära talet 0010 0010.

- w innehåller
- a) 0x20
 - b) 18 (decimalt)
 - c) 0xEE
 - d) 0xFC

6.2 PortC har insignalen 0x34. Vad blir då utsignalen till PortB efter det att följande program körts?

```
MOVWF PORTC,W
IORLW 0XF0
MOVWF TEMP
RLF TEMP
MOVWF TEMP,W
MOVWF PORTB
```

6.3 Ange vilken/vilka bit/ar av C och Z i statusregistret STATUS som sätts då följande programsekvens utförs. Anta att bitarna är nollställda från början.

- ```
MOVLW 0X3F
ADDLW 0XC1
```
- a) ingen bit
  - b) Z-biten
  - c) C-biten
  - d) Z-biten och C-biten

6.4 Vilken instruktion ettställer de två MSB i arbetsregistret w utan att övriga bitar påverkas?

- a) IORLW 0XF0
- b) IORLW 0X0F
- c) ANDLW 0XE0
- d) ANDLW 0X0E
- e) IORLW 0XC0
- f) ANDLW 0XC0
- g) IORLW 0X3F
- h) ANDLW 0X3F

6.5 Vilket minne används för att lagra program i PIC 16F877A?

- a) DRAM
- b) SRAM
- c) FLASH
- d) EEPROM

**Uppgift 7 (3p)**

Bilden nedan visar en ström av bitar som skickas med seriell kommunikation i pilens riktning (biten längst till höger når mottagaren först). Själva värdet (ASCII-tecknet) är 7 bitar och till det kommer startbit, stoppbit och paritetsbit (udda paritet). Med hjälp av paritetsbiten kan mottagaren se om värdet har överförts korrekt

1 1 1 0 1 1 1 0 1 0 0 →

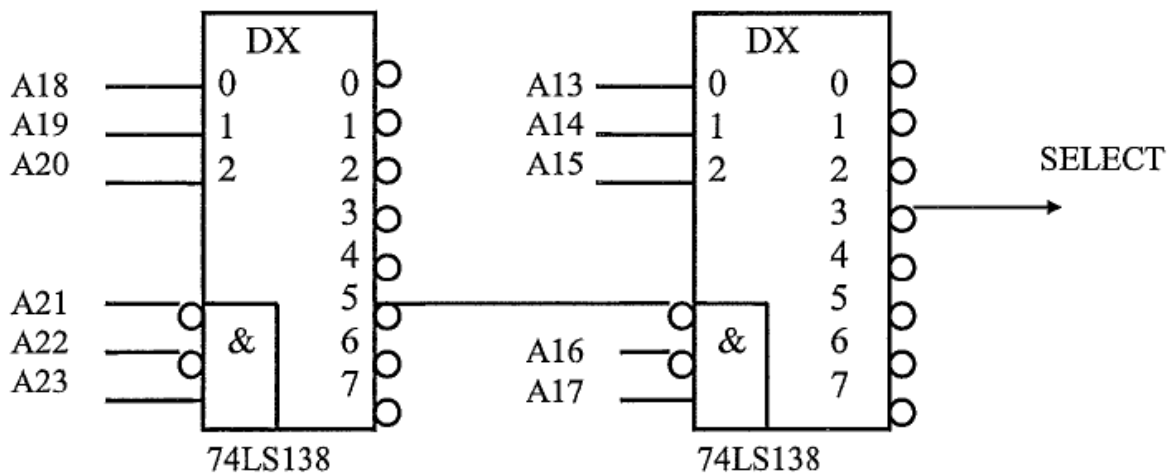
- a) Har värdet överförts korrekt? Motivera väl!
- b) Vilket tecken är det som överförts? Se bilaga.

**Uppgift 8. (2 p)**

En microcontroller kan byggas upp på olika sätt. Vad innebär det att en microcontroller är uppbyggd enligt Harvard-arkitektur resp. von Neuman-arkitektur?  
Förklara kortfattat skillnaden.

**Uppgift 9. (3p)**

Vilket adressrum gäller för den minneskrets som ansluts till kontakten SELECT?  
Svara med lägsta och högsta adress och adressrummets storlek (hexadecimal form).



**Uppgift 10 (2p)**

Beskriv ingående begreppet handskakning, ange när det används och hur det går till.  
Rita gärna figur.

**Uppgift 11 (3p)**

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 12 (2p)**

Skriv en subrutin, som avgör om ett tal som finns lagrat i filregister TMP är jämnt eller udda. Om talet är udda ska innehållet i TMP vara oförändrat, om talet är jämnt ska 0 placeras i TMP.

### Uppgift 13 (4p)

Skriv ett väl kommenterat program i PIC-assembler, som bestämmer hur många bitar i minnescell 0x20 (INDATA) som antar värdet 1. Placera resultatet i minnescell 0x21 (UTDATA).

### Uppgift 14 (4p)

a) Vilket av följande påståenden är sant om nedanstående subrutin:

|         |        |          |                                                                                   |
|---------|--------|----------|-----------------------------------------------------------------------------------|
| COMPARE | MOVF   | 0x20,0   | Ledning: subwf f,d ger<br>C=1 om differensen $\geq$ 0<br>C=0 om differensen $<$ 0 |
|         | SUBWF  | 0x21,0   |                                                                                   |
|         | BTFSS  | STATUS,C |                                                                                   |
|         | CALL   | SUBRUT   |                                                                                   |
|         | RETURN |          |                                                                                   |

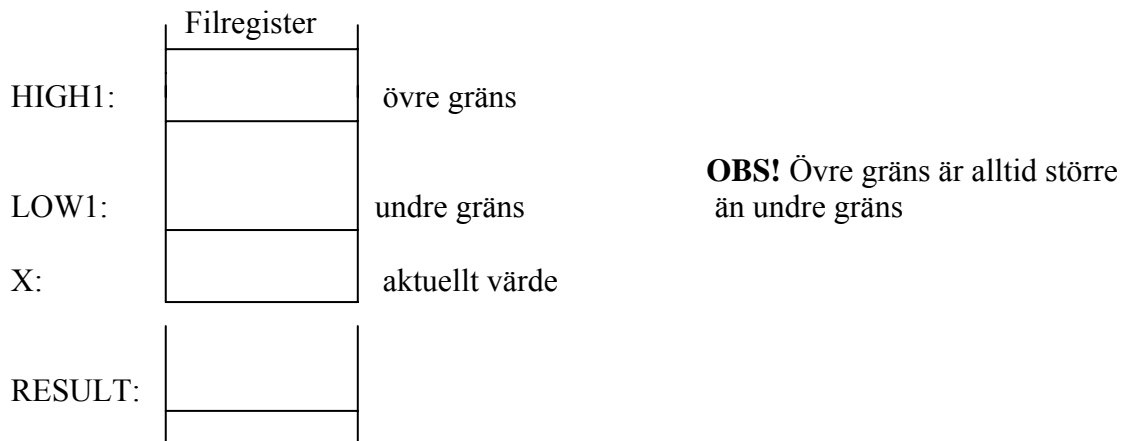
1. om innehållet på minnesadressen 0x20 > innehållet på minnesadressen 0x21 anropas subrutinen subrut
2. om innehållet på minnesadressen 0x20 > innehållet på minnesadressen 0x21 anropas inte subrutinen subrut
3. om innehållet på minnesadressen 0x20 > värdet 0x21 anropas subrutinen subrut
4. om hexadecimala värdet 0x20 < 0x21 anropas subrutinen subrut

b) Skriv ett program som räknar antalet negativa tal (MSB = 1) i tabellen TABELL t ex 0XD7, 0XF3, 0X1A, 0XA7, 0X8D, 0X13, 0XCA och 0X27.  
Placera antalet negativa tal på adressen NEG i minnet.

|        |       |       |                  |
|--------|-------|-------|------------------|
| TABELL | ADDWF | PCL,1 | ;(PC)+(W)-->(PC) |
|        | RETLW | 0XD7  | ;NR 0            |
|        | RETLW | 0XF3  | ;NR 1            |
|        | RETLW | 0X1A  | ;NR 2            |
|        | RETLW | 0X0A  | ;NR 3            |
|        | RETLW | 0X08  | ;NR 4            |
|        | RETLW | 0X13  | ;NR 5            |
|        | RETLW | 0XCA  | ;NR 6            |
|        | RETLW | 0X27  | ;NR 7            |

### Uppgift 15 (7p)

- a) Skriv en SUBROUTIN med namnet JFR i assemblerspråket för PIC.  
Subrutinen skall jämföra innehållet i filregistret X med två värden i filregistret enligt:



Filregistret RESULT ska avspegla resultatet av jämförelsen enligt:

|         |   |   |   |   |   |   |   |   |
|---------|---|---|---|---|---|---|---|---|
|         | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESULT: | X | X |   |   |   |   |   |   |

bit 7 = 1 om undre gräns < (X) ≤ övre gräns  
0 annars

bit 6 = 1 om (X) ≤ undre gräns  
0 om (X) > övre gräns

bit 5 till bit 0 spelar ingen roll

Dvs - bit 7 = 1 anger att (X) är INOM gränser och då har bit 6 ingen betydelse.  
- bit 7 = 0 anger att (X) är UTANFÖR gränser och då avspeglar bit 6 hur.

- b) Skriv ett assemblerprogram MAIN som använder subrutinen JFR i a) för att jämföra ett 8-bitars värde (t ex en temperatur) som finns lagrat i filregistret på adressen X mot en övre gräns (för hög temperatur) och en undre gräns (för låg temperatur).

Värdet för övre respektive undre gräns finns lagrat i minnesorden med adresserna HIGH1 respektive LOW1 (se ovan).

Beroende på resultatet av jämförelsen ska bit 3 i PORTB (värme till/från) påverkas enligt:

|                              |   |                                |
|------------------------------|---|--------------------------------|
| - värdet > övre gräns        | → | bit 3 nollställes (värme från) |
| - värdet ≤ undre gräns       | → | bit 3 ettställes (värme till)  |
| - värdet ligger inom gränser | → | bit 3 oförändrad               |

OBS! Övriga bitar i utporten får ej påverkas!

| ASCII-koden |         |       |         |             | Appendix D |         |       |         |             |        |         |       |         |             |
|-------------|---------|-------|---------|-------------|------------|---------|-------|---------|-------------|--------|---------|-------|---------|-------------|
| Tecken      | Binär   | Oktal | Decimal | Hexadecimal | Tecken     | Binär   | Oktal | Decimal | Hexadecimal | Tecken | Binär   | Oktal | Decimal | Hexadecimal |
| NUL         | 0000000 | 000   | 000     | 00          | +          | 0101011 | 053   | 043     | 2B          | V      | 1010110 | 126   | 086     | 56          |
| SOH         | 0000001 | 001   | 001     | 01          | ,          | 0101100 | 054   | 044     | 2C          | W      | 1010111 | 127   | 087     | 57          |
| STX         | 0000010 | 002   | 002     | 02          | -          | 0101101 | 055   | 045     | 2D          | X      | 1011000 | 130   | 088     | 58          |
| ETX         | 0000011 | 003   | 003     | 03          | .          | 0101110 | 056   | 046     | 2E          | Y      | 1011001 | 131   | 089     | 59          |
| EOF         | 0000100 | 004   | 004     | 04          | /          | 0101111 | 057   | 047     | 2F          | Z      | 1011010 | 132   | 090     | 5A          |
| ENQ         | 0000101 | 005   | 005     | 05          | 0          | 0110000 | 060   | 048     | 30          | Å      | 1011011 | 133   | 091     | 5B          |
| ACK         | 0000110 | 006   | 006     | 06          | 1          | 0110001 | 061   | 049     | 31          | Ö      | 1011100 | 134   | 092     | 5C          |
| BEL         | 0000111 | 007   | 007     | 07          | 2          | 0110010 | 062   | 050     | 32          | Ä      | 1011101 | 135   | 093     | 5D          |
| BS          | 0001000 | 010   | 008     | 08          | 3          | 0110011 | 063   | 051     | 33          | U      | 1011110 | 136   | 094     | 5E          |
| HT          | 0001001 | 011   | 009     | 09          | 4          | 0110100 | 064   | 052     | 34          |        | 1011111 | 137   | 095     | 5F          |
| LF          | 0001010 | 012   | 010     | 0A          | 5          | 0110101 | 065   | 053     | 35          | ˆ      | 1100000 | 140   | 096     | 60          |
| VT          | 0001011 | 013   | 011     | 0B          | 6          | 0110110 | 066   | 054     | 36          | a      | 1100001 | 141   | 097     | 61          |
| FF          | 0001100 | 014   | 012     | 0C          | 7          | 0110111 | 067   | 055     | 37          | b      | 1100010 | 142   | 098     | 62          |
| CR          | 0001101 | 015   | 013     | 0D          | 8          | 0111000 | 070   | 056     | 38          | c      | 1100011 | 143   | 099     | 63          |
| SO          | 0001110 | 016   | 014     | 0E          | 9          | 0111001 | 071   | 057     | 39          | d      | 1100100 | 144   | 100     | 64          |
| SI          | 0001111 | 017   | 015     | 0F          | :          | 0111010 | 072   | 058     | 3A          | e      | 1100101 | 145   | 101     | 65          |
| DLE         | 0010000 | 020   | 016     | 10          | ;          | 0111011 | 073   | 059     | 3B          | f      | 1100110 | 146   | 102     | 66          |
| DC1         | 0010001 | 021   | 017     | 11          | <          | 0111100 | 074   | 060     | 3C          | g      | 1100111 | 147   | 103     | 67          |
| DC2         | 0010010 | 022   | 018     | 12          | =          | 0111101 | 075   | 061     | 3D          | h      | 1101000 | 150   | 104     | 68          |
| DC3         | 0010011 | 023   | 019     | 13          | >          | 0111110 | 076   | 062     | 3E          | i      | 1101001 | 151   | 105     | 69          |
| DC4         | 0010100 | 024   | 020     | 14          | ?          | 0111111 | 077   | 063     | 3F          | j      | 1101010 | 152   | 106     | 6A          |
| NAK         | 0010101 | 025   | 021     | 15          | É          | 1000000 | 100   | 064     | 40          | k      | 1101011 | 153   | 107     | 6B          |
| SYN         | 0010110 | 026   | 022     | 16          | A          | 1000001 | 101   | 065     | 41          | l      | 1101100 | 154   | 108     | 6C          |
| ETB         | 0010111 | 027   | 023     | 17          | B          | 1000010 | 102   | 066     | 42          | m      | 1101101 | 155   | 109     | 6D          |
| CAN         | 0011000 | 030   | 024     | 18          | C          | 1000011 | 103   | 067     | 43          | n      | 1101110 | 156   | 110     | 6E          |
| EM          | 0011001 | 031   | 025     | 19          | D          | 1000100 | 104   | 068     | 44          | o      | 1101111 | 157   | 111     | 6F          |
| SUB         | 0011010 | 032   | 026     | 1A          | E          | 1000101 | 105   | 069     | 45          | p      | 1110000 | 160   | 112     | 70          |
| ESC         | 0011011 | 033   | 027     | 1B          | F          | 1000110 | 106   | 070     | 46          | q      | 1110001 | 161   | 113     | 71          |
| FS          | 0011100 | 034   | 028     | 1C          | G          | 1000111 | 107   | 071     | 47          | r      | 1110010 | 162   | 114     | 72          |
| GS          | 0011101 | 035   | 029     | 1D          | H          | 1001000 | 110   | 072     | 48          | s      | 1110011 | 163   | 115     | 73          |
| RS          | 0011110 | 036   | 030     | 1E          | I          | 1001001 | 111   | 073     | 49          | t      | 1110100 | 164   | 116     | 74          |
| US          | 0011111 | 037   | 031     | 1F          | J          | 1001010 | 112   | 074     | 4A          | u      | 1110101 | 165   | 117     | 75          |
| SP          | 0100000 | 040   | 032     | 20          | K          | 1001011 | 113   | 075     | 4B          | v      | 1110110 | 166   | 118     | 76          |
| !           | 0100001 | 041   | 033     | 21          | L          | 1001100 | 114   | 076     | 4C          | w      | 1110111 | 167   | 119     | 77          |
| "           | 0100010 | 042   | 034     | 22          | M          | 1001101 | 115   | 077     | 4D          | x      | 1111000 | 170   | 120     | 78          |
| #           | 0100011 | 043   | 035     | 23          | N          | 1001110 | 116   | 078     | 4E          | y      | 1111001 | 171   | 121     | 79          |
| ¤           | 0100100 | 044   | 036     | 24          | O          | 1001111 | 117   | 079     | 4F          | z      | 1111010 | 172   | 122     | 7A          |
| ¥           | 0100101 | 045   | 037     | 25          | P          | 1010000 | 120   | 080     | 50          | ä      | 1111011 | 173   | 123     | 7B          |
| &           | 0100110 | 046   | 038     | 26          | Q          | 1010001 | 121   | 081     | 51          | ö      | 1111100 | 174   | 124     | 7C          |
| '           | 0100111 | 047   | 039     | 27          | R          | 1010010 | 122   | 082     | 52          | å      | 1111101 | 175   | 125     | 7D          |
| (           | 0101000 | 050   | 040     | 28          | S          | 1010011 | 123   | 083     | 53          | ü      | 1111110 | 176   | 126     | 7E          |
| )           | 0101001 | 051   | 041     | 29          | T          | 1010100 | 124   | 084     | 54          | DEL    | 1111111 | 177   | 127     | 7F          |
| *           | 0101010 | 052   | 042     | 2A          | U          | 1010101 | 125   | 085     | 55          |        |         |       |         |             |

Fig. D-1 7 bitars ASCII-kod enligt svensk standard.

**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)
- Bit5-6: IRP biten används inte på PIC16F8X och bör vara satt till noll för att garantera framåtkompatibilitet med större PIC'ar.  
RP1:RP0: Register Bank Select bit (används för direkt adressering).  
00=Bank 0 (00j-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 PIC16F8X. 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 SLEEP 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 har blivit noll.
- Bit1: DC: Digit carry/borrow' bit (för ADDWF och ADDLW instruktioner)(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(låne-bit) är förhållandena omkastade