Lösningar och svar

1.

a) Man kan förmoda att CS_1 och CS_2 används till minnesmoduler eftersom de vardera tar upp stora adressområden. CS_1 används säkerligen till ett minne som har en R/W-ingång, d v s ett RWM. CS_2 används till ett minne som ej har någon R/W-ingång, d v s ett ROM.

Då CS_3 och CS_4 tar upp en adress vardera så kan man förmoda att de används till I/O-portar. CS_3 är aktiv när R/W är 0, d v s under skrivning och bör därför gå till en utport. CS_4 är aktiv när R/W är 1, d v s under läsning och bör således gå till en inport.

b) Minnesmodulerna i adressrum M tar upp följande adressområden:

	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	
RWM:	1	1	0	x	x	x	x	x	x	x	x	x	x	x	x	x	C000h - DFFFh
ROM:	1	1	1	1	x	x	x	x	x	x	x	x	x	x	x	x	F000h - FFFFh

In- och utportarna i adressrum M tar upp följande adressområden:

	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	
Utport:	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	0	60C0h
Inport:	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	1	60C1h

c) Processortillverkaren garanterar att adress- och R/W-signalerna är stabila och giltiga under den tid som signalen E är 1. Därför använder man signalen E som en VMA- (Valid Memory Adress) signal.

---

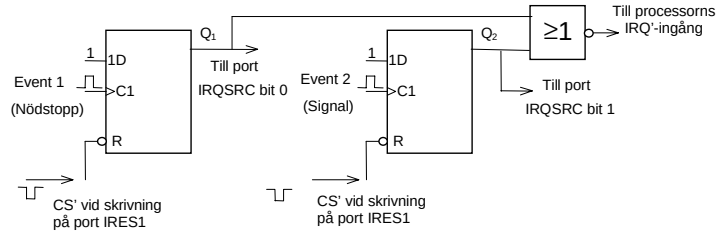
2. **a)** Minnesinnehåll efter assemblering och laddning:

```
3250 BF                 Number   FCB   %10111111
3251                    Portcopy RMB   1
3252 B6 32 50                    LDAA  Number
3255 16 37 20                    JSR   KLIR
3258 7B 04 D0                    STAB  Port
325B 7B 32 51                    STAB  Portcopy
```

b) Man startar exekveringen med g 3252, ty där börjar maskininstruktionerna.

3.

a) Koppling



b)

```
*********************************************************************
*SUBRUTIN-IRQINIT
*Beskrivning:  Rutinen nollställer D-vipporna, lägger in adressen
*              till avbrottsrutinen läggs in på adressen FFF2 och
*              FFF3 samt att I-flaggan nollställs.
*Anrop:        BSR IRQINIT
*Indata:       Inga
*Utdata:       Inga
*Reg-påverkan: Ingen
*Anrop subr:   Ingen
*********************************************************************
IRQINIT   PSHA
          PSHX
          LDX    #IRQR          adressen till avbrottsrutinen läggs
*         STX    $FFF2          på adressen FFF2 och FFF3
          LDAA   IRES1          nollställer D-vippa 1
          LDAA   IRES2          nollställer D-vippa 2
          ANDCC  #%11101111     I-flaggan nollställs
          PULX
          PULA
          RTS
*********************************************************************
```

c)

```
*********************************************************************
*AVBROTTSRUTIN-IRQR
*Beskrivning:  Om swith 1 har aktiverats, dvs BITPOS 7=1 så görs
*              nödstopp. Om swith 2 har aktiverats, dvs
*              BITPOS 6=1 så körs larmrutinen.
*Anrop:        via IRQ
*Indata:       Inga
*Utdata:       Inga
*Reg-påverkan: CC
*Anrop subr:   SIGNAL
*********************************************************************
IRQR      LDAA   IRQSRG        hämtar vippornas status
          ASLA                 shiftar in BITPOS 7 i c-flaggan
          BCC    NO_STOP       undersöker statusbit för nödstopp
          LDAB   IRES1         nollställer vippa 1
          BRA    DRPGM         om c=1 hoppa till DRPGM
NO_STOP   ASLA                 shiftar in BITPOS 6 i c-flaggan
          BCC    NO_SIGNAL     undersöker statusbit för larm
          LDAB   IRES2         nollställer vippa 2
          ANDCC  #%11101111    nollställer I-flaggan
          BSR    SIGNAL        om c=1 hoppa till subrutinen SIGNAL
NO_SIGNAL RTI
*********************************************************************
```

4.

a) Första samplet tas så snart startbiten upptäcks. Eftersom mottagaren undersöker den inkomna signalen med frkvensen 160 kHz, så kan samplingsögonblicket ε ligga inom intervallet 0 - 6,25 µs.

b) Man kan misstänka att felet beror på att mottagarens frekvens skiljer sig från sändarens. Ramfel ("framing error") erhålls när den sända paritetsbiten tolkas som stoppbit och det sker när mottagaren läser stoppbiten för tidigt. Mottagarens frekvens är då högre än sändarens.

c) Man upptäcker detta när den sända paritetsbiten är 0. Paritetsfelet beror på att mottagaren dessutom läser paritetsbiten så tidigt att den egentligen läser den sända bit 6 två gånger. Andra gången förväntar sig mottagaren en paritetsbit. Paritetsfel signaleras om sänd bit 7 och sänd paritetsbit har olika värden.

5.

```
// Filen ports.h
#define ML5DISPL_SEGMENT_ADR    0xC02
#define ML5DISPL_DIGITS_ADR     0xC03
#define ML5DISPL_SEGMENT        *((portptr) ML5DISPL_SEGMENT_ADR)
#define ML5DISPL_DIGITS         *((portptr) ML5DISPL_DIGITS_ADR)

// Filen displayML5.c
void display_hex(unsigned int number);

// Filen displayML5.c
#include "displayML5.h"
#include "ports.h"

void display_hex(unsigned int number) {
  static unsigned char tab[16] =  {0x77, 0x24, 0x5d, 0x6d,
                                   0x2e, 0x6b, 0x7b, 0x25,
                                   0x7f, 0x6f, 0x3f, 0x7a,
                                   0x58, 0x7c, 0x5b, 0x1b};
  int digNo, radbits;
  for (digNo=0, radbits=0x1; digNo<6; digNo++, radbits<<=1) {
    ML5DISPL_DIGITS = ~radbits;                 // markera indikator som skall aktiveras
    ML5DISPL_SEGMENT = tab[number & 0xf];       // ange segment
    number >>= 4;
  }
}
```

6.

```
int strcmp(char *s, char *t) {
  for ( ; *s == *t; s++, t++)
    if (*s == '\0')
      return 0;
  return *s - *t;
}
```

---

7. Svaren kommer!

a)

b)