



TENTAMEN

KURSNAMN	Maskinorienterad programmering
PROGRAM:	Dataingenjör och elektroingenjör åk 1/ lp 3 Mekatronikingenjör åk 2/ lp 3
KURSBETECKNING	LEU500
EXAMINATOR	Lars-Eric Arebrink
TID FÖR TENTAMEN	Måndag 2010-08-23 kl 14.00 – 18.00
HJÄLPMEDEL	Av institutionen utgiven ”Instruktionslista för CPU12” (INS2) Tabellverk eller miniräknare får ej användas.
ANSV LÄRARE: besöker tentamen	Lars-Eric Arebrink 772 5718 Vid flera tillfällen
DATUM FÖR ANSLAG av resultat samt av tid och plats för granskning	Resultatlistor med de anonyma koderna anslås senast 2010-09-06 på kursens hemsida. Granskning på institutionen 2010-09-06 och 2010-09-08 kl 12.30-13.00.
ÖVRIG INFORM. BETYGSGRÄNSER. SLUTBETYG	Tentamen omfattar totalt 60 poäng. 24p ≤ betyg 3 < 36 p ≤ betyg 4 < 48 p ≤ betyg 5 För godkänt slutbetyg 3, 4 eller 5 på kursen fordras betyg 3, 4 eller 5 på tentamen samt godkända laborationer.

1. Besvara kortfattat följande frågor, som alla utom i) - l) avser CPU12.

- a) Översätt assemblerinstruktionen DAA till maskinspråk och visa hur maskinkoden placeras i minnet. **(1p)**
- b) Vilken assemblerinstruktion har den hexadecimala maskinkoden 18 03 12 34 56 78? **(2p)**
- c) Assemblerinstruktionen EXG D,X kan skrivas som en alternativ assemblerinstruktion. Vilken? **(1p)**
- d) Översätt assemblerinstruktionen DEC -\$100,Y till maskinspråk. Visa hur maskinkoden placeras i minnet. **(2p)**
- e) Assemblerinstruktionen LBRN \$1000 har operationskoden på adressen 1088H. Visa hur maskinkoden placeras i minnet. På vilken adress utförs nästa instruktion? **(2p)**
- f) Förklara vad H -flaggan i CC -registret används till. **(2p)**

För vilka värden W ($0 \leq W \leq 255$) på minnesordet på adressen Wadr utförs hoppet i g) och h)?

- g)

g)	LDAB	# $\$20$			
	CMPB	Wadr			
	BLE	Hopp			(2p)
- h)

h)	LDAB	# $\$78$			
	CMPB	Wadr	(Tänk på att overflow kan inträffa vid jämförelsen!)		
	BMI	Hopp			(3p)
- i) När är det lämpligt att använda asynkron resp. synkron seriekommunikation? **(2p)**
- j) Bitar som sänds på en CAN-buss är okodade. Hur kan synkroniseringen upprätthållas om ett stort antal lika bitar sänds i sekvens? **(1p)**
- k) Skriv det decimala talet -378,625 som ett 32-bitars flyttal enligt IEEE-standard 754-1985 (23 bitar av mantissan och 8 bitars karakteristika). Ge svaret på hexadecimal form. **(2p)**
- l) Visa approximativt vilket talområde (decimalt) man får om man använder ett flyttalsformat med 6 bitar i exponenten och man antar att alla exponentvärden kan användas. **(2p)**

2.

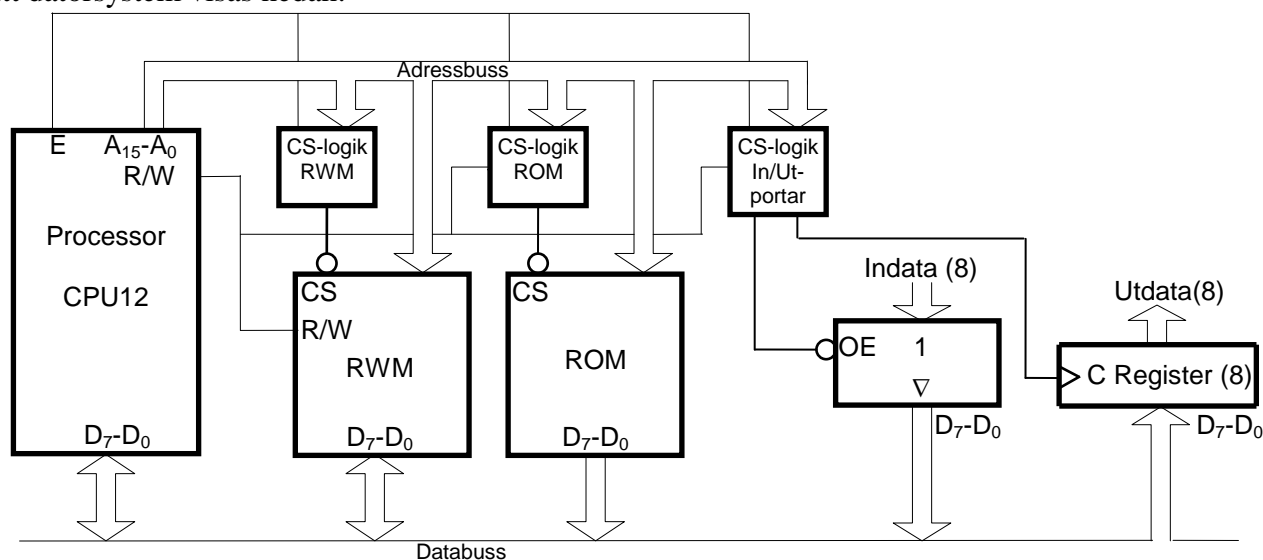
- a) Skriv en subrutin SWAP i assemblerspråk för CPU12, som byter plats mellan den vänstra halvan och den högra halvan av alla 8-bitars dataord i en nollterminerad sträng i minnet. Vid anrop av subrutinen skall adressen till strängen finnas i X-registret. Endast flaggregistret får vara förändrat vid återhopp. För full poäng skall programmet vara ”korrekt” radkommenterat. **(5p)**
- b) Programsekvensen nedan visar ett sätt att överföra tre inparametrar till en subrutin. Visa med en programsekvens hur subrutinen kan hämta inparameter 1 till register A, inparameter 2 till register X och inparameter 3 till register Y och senare göra ett återhopp till adressen efter andra inparametern nedan. Antag att inga registerinnehåll sparas på stacken. Tänk på att subrutinen kan anropas från flera ställen i samma huvudprogram.

```

      .
      .
      JSR      SUBRUT
      FCB      $12          Inparameter 1
      FDB      $3456       Inparameter 2
      FDB      $789A       Inparameter 3
      .
      .
  
```

(4p)

3. Ett datorsystem visas nedan:



Figuren ovan visar principen för anslutning av externa minnesmoduler och externa in-/utportar till processorn CPU12.

En 4 kbyte RWM-modul skall vara placerad med start på adress F000H.

På adresserna FFF0H-FFFFH skall en 16 byte ROM-modul vara placerad.

På adressen 200H skall en inport och en utport vara placerad.

Resten av adressrummet skall fyllas ut med en 64 kbyte ROM-modul, men 512 adresser med start på adress 0 skall inte aktivera någon minnesmodul eller port vid läsning eller skrivning.

Det innebär att inporten skall prioriteras och att RWM-modulen skall prioriteras före 64kbyte ROM-modulen samt att 16 byte ROM-modulen skall prioriteras före övriga minnesmoduler.

Rita CS-logiken för minnesmodulerna och portarna. Ange adressintervallen för alla modulerna.

Använd fullständig adressavkodning. Endast grundläggande logikgrindar med valfritt antal ingångar får användas. **(8p)**

4. En dator med processorn CPU12 skall användas i en tillämpning där två sensorer, Sense1 och Sense2, skall läsas av via IRQ-avbrott. Sense1 skall läsas av tjugo gånger per sekund och Sense2 en gång per minut. Databitarna (8 st) från sensorerna är anslutna till var sin inport på de symboliska adresserna PORT1 och PORT2.

Det finns en binär signal med den konstanta frekvensen 100 Hz tillgänglig för generering av avbrott och en ledig inport finns på adressen 410H. På adresserna 418H-41FH finns inget minne eller andra enheter anslutna och det finns inga fler avbrottskällor.

- a) Föreslå en koppling för avbrottsgenerering på processorns IRQ'-ingång. D-vippor, NAND- och NOT-grindar får användas. **(2p)**
- b) Skriv en avbrottsrutin IRQRUT, som läser av sensorerna enligt beskrivningen ovan och placerar de avlästa värdena på de symboliska adresserna SE1 och SE2 i minnet. Ledigt utrymme för globala variabler finns på adresserna 1880H-188FH. **(4p)**
- c) Skriv ett avsnitt av huvudprogrammet där IRQ-avbrott initieras. IRQ-vektorn antas vara placerad i RWM på adresserna FFF2H och FFF3H. Avbrottsrutinen börjar på adressen IRQRUT. **(3p)**

Alla odefinierade symboliska adresser ovan är definierade på annat ställe i programmet. Assemblerspråk för processorn CPU12 skall användas. Radkommentarer skall finnas!

5.

- a) Du använder en korskompilator för HCS12 med följande anropskonventioner för C-funktioner:

- Parameterlistan behandlas från höger till vänster. Samtliga inparametrar överförs via processorns stack.
- Lokala variabler behandlas i den ordning de deklarerats, dvs sist behandlad finns överst i stacken.
- Varje funktion som har lokala variabler inleds med prologen `LEAS -?,SP` och avslutas med epilogen `LEAS ?,SP` följt av `RTS`.
- Returparameter lämnas i D- eller B-registret beroende på storlek.

Antag att en funktion definieras på följande sätt:

```
int funca( int a, long b, short c )
{
    char d;
    int e
    short f;
    . . . .
}
```

Visa stackens innehåll direkt efter det att funktionens prolog har körts. Platsen för samtliga variabler skall visas. **(2p)**

5. (forts.)

- b)** Översätt C-funktionen nedan till assemblyspråk för CPU12:
(Antag att anropskonventionerna i a-uppgiften gäller)

```
int funcb(char n){  
    char i;  
    int j;  
    j = 0;  
    for ( i = 0; i < n; i++)  
        j = j + n;  
    return j;  
}
```

(5p)**6.**

- a)** Metoderna som används för att förbättra minnesprestanda hos primärminnet i en dator bygger på egenskaper hos minnesmodulerna (halvledarkretsar) och på hur programmen använder minnet samt hur data lagras i minnet. Vilka är dessa egenskaper? Hur kan man utnyttja dem? **(3p)**
- b)** Förklara varför man inte kan använda ”flash”-minnen som vanliga RWM i ett datorsystem. **(2p)**

Assemblerspråket för CPU12 .

Assemblerspråket använder sig av mnemoniska beteckningar som processorkonstruktören MOTOROLA specificerat för maskininstruktioner och instruktioner till assemblern, s k pseudoinstruktioner eller assemblerdirektiv. Pseudoinstruktionerna listas i tabell 1.

Tabell 1

Direktiv	Förklaring
ORG N	Placerar den efterföljande koden med början på adress N. (ORG för ORiGin = ursprung)
L RMB N	Avsätter N bytes i följd i minnet (utan att ge dem värden), så att programmet kan använda dem. Följden placeras med början på adressen L. (RMB för Reseve Memory Bytes)
L EQU N	Ger symbolen L konstantvärdet N. (EQU för EQUates = beräknas till)
L FCB N1, N2	Avsätter en byte för varje argument i följd i minnet. Respektive byte ges konstantvärdet N1, N2 etc. Följden placeras med början på adressen L. (FCB för Form Constant Byte)
L FDB N1, N2	Avsätter ett bytepar (två bytes) för varje argument i följd i minnet med mest signifikant byte på den lägsta adressen. Respektive bytepar ges konstantvärdet N1, N2 etc. Följden placeras med början på adressen L. (FDB för Form Double Byte)
L FCS "ABC"	Avsätter en byte för varje tecken i teckensträngen "ABC" i följd i minnet. Respektive byte ges ASCII-värdet för A B C, etc. Följden placeras med början på adressen L. (FCS för Form Character String)

ASCII-koden

Tabell 2 7-bitars ASCII

0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1	b ₆ b ₅ b ₄ b ₃ b ₂ b ₁ b ₀
NUL	DLE	SP	0	@	P	`	p	0 0 0 0
SOH	DC1	!	1	A	Q	a	q	0 0 0 1
STX	DC2	"	2	B	R	b	r	0 0 1 0
ETX	DC3	#	3	C	S	c	s	0 0 1 1
EOT	DC4	\$	4	D	T	d	t	0 1 0 0
ENQ	NAK	%	5	E	U	e	u	0 1 0 1
ACK	SYN	&	6	F	V	f	v	0 1 1 0
BEL	ETB	'	7	G	W	g	w	0 1 1 1
BS	CAN	(8	H	X	h	x	1 0 0 0
HT	EM)	9	I	Y	i	y	1 0 0 1
LF	SUB	*	:	J	Z	j	z	1 0 1 0
VT	ESC	+	;	K	[Ä	k	{ä	1 0 1 1
FF	FS	,	<	L	\Ö	l	ö	1 1 0 0
CR	GS	-	=	M]Å	m	}å	1 1 0 1
S0	RS	.	>	N	^	n	~	1 1 1 0
S1	US	/	?	O	_	o	RUBOUT (DEL)	1 1 1 1