



## TENTAMEN

<b>KURSNAMN</b>	<b>Maskinorienterad programmering</b>
<b>PROGRAM:</b>	<b>Dataingenjör och elektroingenjör åk 1/ lp 3 Mekatronikingenjör åk 2/ lp 3</b>
<b>KURSBETECKNING</b>	<b>LEU500</b>
<b>EXAMINATOR</b>	<b>Lars-Eric Arebrink</b>
<b>TID FÖR TENTAMEN</b>	<b>Lördag 2011-01-15 kl 8.30 – 12.30</b>
<b>HJÄLPMEDEL</b>	<b>Av institutionen utgiven ”Instruktionslista för CPU12” (INS2) Tabellverk eller miniräknare får ej användas.</b>
<b>ANSV LÄRARE:</b>  besöker tentamen	<b>Lars-Eric Arebrink 772 5718 Vid flera tillfällen</b>
<b>DATUM FÖR ANSLAG</b> av resultat samt av tid och plats för granskning	<b>Resultatlistor anslås senast 2011-02-03 på kursens hemsida. Granskning på institutionen 2011-02-03 och 2011-02-04 kl 12.30-13.00.</b>
<b>ÖVRIG INFORM.</b> <b>BETYGSGRÄNSER.</b> <b>SLUTBETYG</b>	<b>Tentamen omfattar totalt 60 poäng. 24p ≤ betyg 3 &lt; 36 p ≤ betyg 4 &lt; 48 p ≤ betyg 5 För godkänt slutbetyg 3, 4 eller 5 på kursen fordras betyg 3, 4 eller 5 på tentamen samt godkända laborationer.</b>

1. Besvara kortfattat följande frågor, som alla utom h) - j) avser CPU12.

- a) Assemblerinstruktionen ANDCC #\$FE kan skrivas på ett alternativt sätt. Visa detta sätt. **(2p)**
- b) Vilken assemblerinstruktion har den hexadecimala maskinkoden 18 09 5A 15 38? **(2p)**
- c) Översätt assemblerinstruktionen ORAB -100,SP till maskinspråk. Visa hur maskinkoden placeras i minnet. **(2p)**
- d) Översätt assemblerinstruktionen IBNE D,\$1742 till maskinspråk. Operationskoden finns på adressen 1800<sub>16</sub>. Visa hur maskinkoden placeras i minnet. **(2p)**

För vilka värden  $W$  ( $0 \leq W \leq 255$ ) utförs hoppet i programavsnitten e) och f)?

- e)
 

e)	LDAA	#\$63		
	CMPA	#W		
	BHI	Hopp		<b>(2p)</b>
- f)
 

f)	LDAB	#\$63		
	CMPB	#W	(Tänk på att "overflow" kan inträffa!)	
	BPL	Hopp		<b>(4p)</b>
- g) Vid kommunikation via en bussledning skickar CAN-noder ibland "stuffbitar". Förklara vad som menas med detta och varför de används. **(2p)**
- h) Skriv talet -200,375 som ett flyttal enligt IEEE-standard 754 (23 bitar av mantissan och 8 bitars karakteristika). **(2p)**
- i) Exponenten i ett flyttal enligt IEEE-standard 754-1985 kodas och placeras på ett speciellt sätt i det "packade" flyttalet. Förklara orsaken till att exponenten kodas och placeras på detta sätt. **(2p)**

2. a) Hur många "E-klockperioder" använder CPU12 (HCS12) för att köra programsekvensen nedan?

```

ORG    $1000
LDX    #TAB
TFR    X,Y
LDD    4,X
YLOOP  LDX    6,Y
XLOOP  DBNE  X,XLOOP
        DBNE  D,YLOOP
        BRA   NEXT
TAB     FDB   15,-7,10,5,20,4,8
NEXT    NOP

```

(4p)

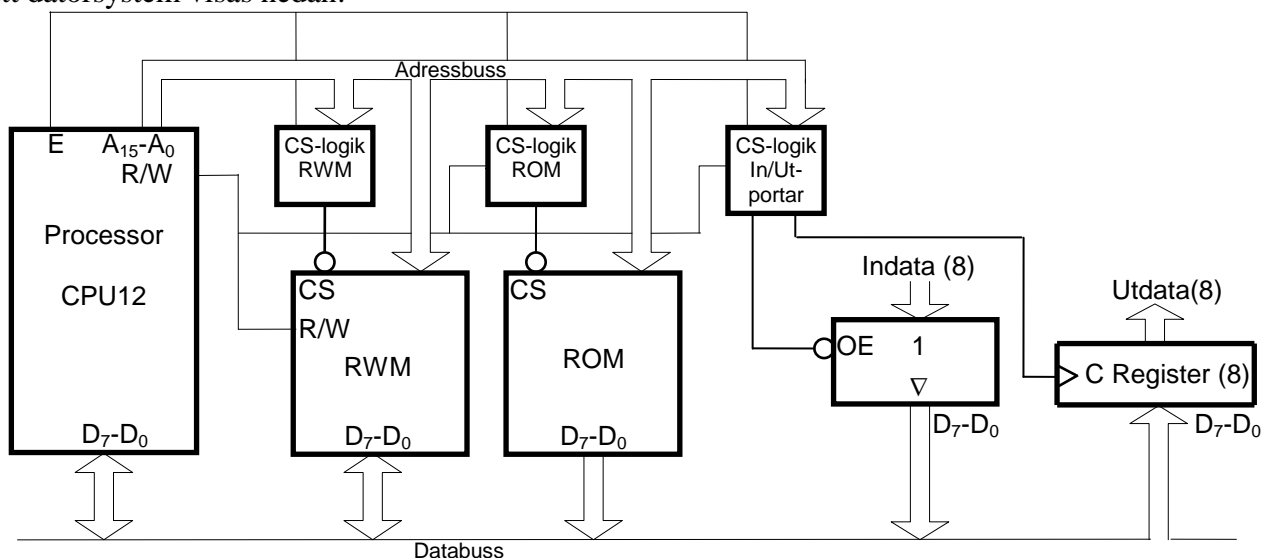
- b) Skriv en subrutin (XCNT) i assemblerspråk för CPU12, som räknar antalet ettor i X-registret och returnerar detta antal i B-registret. Vid återhopp får endast flaggorna och B-registret vara påverkade. Radkommentarer skall finnas.

(4p)

- c) I minnet i en dator med CPU12 finns två lika långa strängar STRING1 och STRING2 med 8-bitars dataord med tecken. Strängarna avslutas med dataordet FF<sub>16</sub>, som bara förekommer som slutord. Skriv en subrutin STRADD i assemblerspråk för CPU12 som adderar varje dataord (ej slutordet) i STRING2 till motsvarande dataord i STRING1 och placerar varje åttabits summa i STRING1. Subrutinen skall dessutom räkna antalet additioner som ger "overflow" och returnera detta antal i D-registret. Vid anrop av STRADD finns adressen till STRING1 i X-registret och adressen till STRING2 i Y-registret. Vid återhopp får endast flaggorna och D-registret vara påverkade. Radkommentarer skall finnas.

(7p)

3. Ett datorsystem visas nedan:



Figuren ovan visar principen för anslutning av externa minnesmoduler och externa in-/utportar till processorn CPU12. En 32 kbyte R/W-modul skall i princip vara placerad från adress 0, men de första 4096 adresserna i adressrummet skall inte aktivera någon minnesmodul eller port vid läsning eller skrivning. En inport och en utport skall också anslutas. De skall ha samma adress och placeras på sista adressen i R/W-modulen. Det innebär att inporten skall prioriteras före R/W-modulen. Resten av adressrummet skall fyllas ut med en 32 kbyte ROM-modul.

Rita CS-logiken för minnesmodulerna och portarna. Använd fullständig adressavkodning. Endast grundläggande logikgrindar med valfritt antal ingångar får användas.

(8p)

4. En dator med processorn CPU12 skall användas i styrenheten i en maskin. Styrprogrammet skall läsa av två givare via IRQ-avbrott, Givare1 tio gånger per sekund och Givare2 en gång per sekund. Databitarna (8 st) från givarna är anslutna till inportar på de symboliska adresserna PORT1 och PORT2.

Det finns en binär signal med den konstanta frekvensen 50 Hz tillgänglig för generering av avbrott och en ledig inport finns på adressen  $810_{16}$ . Adresserna  $818_{16}$ - $81F_{16}$ , som är avsedda för I/O-enheter, används inte av systemet och kan användas vid behov. 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 givarna enligt beskrivningen ovan och placerar de avlästa värdena på de symboliska adresserna GIV1 och GIV2 i minnet. Ledigt utrymme för globala variabler finns på adresserna  $1820_{16}$ - $182F_{16}$ . **(4p)**
- c) Skriv ett avsnitt av huvudprogrammet där IRQ-avbrott initieras. IRQ-vektorn antas vara placerad i RWM på adresserna  $FFF2_{16}$  och  $FFF3_{16}$ . Avbrottsrutinen börjar på adressen IRQRUT. **(2p)**

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

5. I programmeringsspråket C finns den sk switch-satsen med vilken man kan göra flerval. Det innebär att man från ett ställe i programmet kan fortsätta på ett av flera olika ställen. Ett exempel på ett sådant flerval ges nedan.

I en styrenhet för en maskin används CPU12. Ett avsnitt av styrprogrammet skall läsa av värdet SEL (8 bitar), som finns på en inport på den symboliska adressen INPORT, och därefter utföra ett hopp till en av fyra olika adresser, ADR0-ADR3. Vilken av adresserna som väljs bestäms av värdet på variabeln SEL. Om  $SEL = 0$  utförs hopp till ADR0, om  $SEL = 1$  utförs hopp till ADR1 osv. Om  $SEL > 3$  skall hopp ske till den symboliska adressen ERROR. De fyra olika hoppadresserna skall finnas lagrade i minnet i en tabell i styrprogrammet. Vid laddning av programmet i minnet skall också tabellen med hoppadresserna laddas.

Skriv ett huvudprogram som först initierar stackpekaren till  $2000_{16}$  och sedan läser värdet SEL som finns på inporten på adressen INPORT. Om  $SEL < 4$  utförs hopp till en adress enligt beskrivningen i föregående stycke. Om  $SEL > 3$  utförs hopp till adressen ERROR.

Programmet skall skrivas i assemblerspråk för CPU12 och startadressen skall vara  $1000_{16}$ . Programmet skall utformas som flerval och inte som upprepade tvåval.

Subrutinernas startadresser ADR0-ADR3 skall vara  $1100_{16}$ ,  $1150_{16}$ ,  $1200_{16}$ , och  $1380_{16}$ . Radkommentarer skall finnas! **(7p)**

6. Vilken är fördelen och vilken är nackdelen med "direct mapped cache"? **(2p)**

## Bilaga 1

**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 <sub>6</sub> b <sub>5</sub> b <sub>4</sub> b <sub>3</sub> b <sub>2</sub> b <sub>1</sub> b <sub>0</sub>
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