

# ASSEMBLERPROGRAMMERING DEL 1

ÖZGUN MIRTCEV

## INNEHÅLL

Anvisningar för svar i XCEL-dokument	2
1. Addition av två tal med 8 bitars ordlängd	2
2. Addition av två tal med 8 bitars ordlängd	2
3. Klockfrekvens	2
4. Ordlängd	3
5. Hjärnan	3
6. Adressregister X, Y och Z	3
7. RISC & CISC	3
8. Aritmetiska flaggor	4
9. Från byteadress till ordadress	4
10. Sektion för läsminne	4
11. Reservering av dataminne	4
12. Ladda ett register med en konstant	5
13. Ladda ett register från dataminnet	5
14. Spara ett register i dataminnet	5
15. Addition av två register	5
16. Nollställning av en bit i minnet.	6
17. Ett-ställning av en bit	6
18. PORTB	6
19. PORTB	7
20. Bitmask	7
Bilaga A. Svarstyper	8
A.1. Decimalpunkt	9
A.2. Enkla matematiskt uttryck	9
A.3. Enkla differentialekvationsuttryck	9

### Anvisningar för svar i XCEL-dokument

. Följande bör du tänka på då svaren skall anges i ett XCEL-dokument:

- (1) Ändra ALDRIG format (.xls) när du sparar XCEL-dokumentet till något annat (.xlsx, etc.).
- (2) Problem med inmatning av svar till vissa svarsceller där tecken tas bort av kalkylprogrammet.
  - (a) Formatera svarscellerna genom att markera dessa och ändra formatet till TEXT.
  - (b) Ett enkelt och generellt råd är att markera alla celler i svarskolumnen och ändra deras format till TEXT.
  - (c) I OpenOffice använder du höger musknapp för att via kontextmenyn (Format Cells) formatera svarscellerna till formatet TEXT.

#### 1. ADDITION AV TVÅ TAL MED 8 BITARS ORDLÄNGD

*alu\_add\_b.aw2*

Addition av 2 binära tal med 8 bitars ordlängd skall göras.

```

10011111
10101100
+ -----

```

Svaret skall anges som ett 8 bitars tal (R), carryflaggan C samt spillflaggan V.

#### 2. ADDITION AV TVÅ TAL MED 8 BITARS ORDLÄNGD

*alu\_add\_a.aw2*

Addition av 2 binära tal med 8 bitars ordlängd skall göras.

```

00001100
10101100
+ -----

```

Svaret skall anges som ett 8 bitars tal (R), carryflaggan C samt spillflaggan V.

#### 3. KLOCKFREKvens

*cpu\_clock\_cycle\_a.aw2*

En instruktion i AVR-processorn tar en eller två klockcykler att utföra. Om processorn klockas med frekvensen 1 MHz hur lång är då en klockcykel?

## 4. ORDLÄNGD

*cpu\_word\_length\_a.aw2*

Vad menas med ordlängden för en dator?

- a) Antalet dataledningar (bitar) som adressbussen består av.
- b) Bredden uttryckt i antal bitar på det fysiska program- och dataminnet.
- c) Maximala antalet bitar med vilket en beräkningsoperation kan utföras i CPU:n.
- d) Antalet dataledningar (bitar) som databussen består av.

## 5. HJÄRNAN

*cpu\_alu\_a.aw2*

Om ALU:n är hjärtat i en dator som pumpar runt 0:or och 1:or, vad är då hjärnan i ett datorsystem?

- a) Programmeraren.
- b) Styrenheten i datorn.
- c) Programmet.
- d) Programminnet.

## 6. ADRESSREGISTER X, Y OCH Z

*avr\_cpu\_registers\_a.aw2*

Registerparet som betecknas med Y vilka register är det frågan om? Svara på formen R20:R19 med högre registernumret först.

## 7. RISC &amp; CISC

*avr\_risc\_a.aw2*

AVR-processorn har en RISC-arkitektur. Vad är sant (S) respektive falskt (F) av följande påståenden:

- a) RISC står för Reduced Intelligence Super Computer
- b) En RISC-processor har få instruktioner.
- c) En RISC-processor har många instruktioner.
- d) En RISC-processor har en enkel datorarkitektur jämfört med CISC.

## 8. ARITMETISKA FLAGGOR

*avr\_architecture\_a.aw2*

Om vi vill upptäcka overflow i binära heltal med teckenbit vilken flagga i statusregistret SREG skall vi då titta på?

- a) V-flaggan
- b) Z-flaggan
- c) C-flaggan
- d) N-flaggan

## 9. FRÅN BYTEADDRESS TILL ORDADDRESS

*avr\_architecture\_b.aw2*

Då alla instruktioner i AVR-processorn upptar antingen 2 eller 4 byte så är det naturligt att räkna i ord istället för byte. Vår kompilator använder dock byteadresser i listfiler och simulatören i AVRstudio4 använder ordadresser. Detta gör att vi behöver kunna gå mellan byte- och ordadresser. Ett ord är lika med 2 byte. En minnesadress kan då antingen uppges som byteadress eller ordadress. För att få ordadressen delar man byteadressen med två och tar heltalsdelen av detta. Vilken ordadress får vi då byteadressen är 0X0180? Ange svaret som ett fyra siffrors hexadecimalt tal.

## 10. SEKTION FÖR LÄSMINNE

*avr\_intro\_a.aw2*

Vilken sektion återspeglar läsminne:

- a) .text
- b) .data

## 11. RESERVERING AV DATAMINNE

*avr\_intro\_b.aw2*

Antag att adressetiketten VAR har värdet 0x0060. Ange vad som hamnar i minnet från adress 0x0060-0x0067 då följande assemblerdirektiv används:

VAR:

```
.word    -36
.long    166327
.byte    0xA8
.byte    0b11100111
```

Ange det hexadecimala värdena. Observera little-endian-lagring. I svarsddokumentet (EXCEL) anges minnesadresserna som a) till h) där a) är den lägsta minnesadressen och h) den högsta.

## 12. LADDA ETT REGISTER MED EN KONSTANT

*avr\_ldi\_a.aw2*

Skriv instruktionen för att ladda register R21 med konstanten 237.

- a) LDS R21,237
- b) MOV R21,237
- c) LDS 237,R21
- d) LDI R21,237
- e) MOV 237,R21
- f) LDI 237,R21

## 13. LADDA ETT REGISTER FRÅN DATAMINNET

*avr\_lds\_a.aw2*

På adressen 0X00AB i dataminnet finns data som skall laddas till registret R19.

Skriv assemblerkoden för detta. Svara på formen

iii rrr,0Xaaaa

Versaler används för bokstäver.

## 14. SPARA ETT REGISTER I DATAMINNET

*avr\_sts\_a.aw2*

Skriv instruktionen för att spara register R17 till dataminnesadressen 0X00E8.

- a) ST 0X00E8,R17
- b) STS 0X00E8,R17
- c) STS R17,0X00E8
- d) ST Z,R17
- e) ST R17,0X00E8

## 15. ADDITION AV TVÅ REGISTER

*avr\_add\_a.aw2*

Skriv instruktionen för att addera register R20 och R21, resultatet skall hamna i R21. Svaret ges med versaler på formen och med ett blanktecken mellan instruktionen och operanderna (inga blanktecken där).

## 16. NOLLSTÄLLNING AV EN BIT I MINNET.

*avr\_bit\_clear\_a.aw2*

Skriv de assemblerinstruktioner som behövs för att nollställa bit D4 i VAR1 utan att övriga bitar påverkas.

```
.data
.global VAR1
VAR1: .byte 0
```

Bitarna tecknas som D7,D6,...,D0

Använd VERSALER och instruktionerna LDS, STS, ANDI, register R20 och en bitmask på hexadecimal form 0X00-0XFF..

## 17. ETT-STÄLLNING AV EN BIT

*avr\_io\_a.aw2*

Antag att i dataminnet finns en minnescell med den symboliska adressen M:

```
.data
.global M
M: .byte 0
```

Visa hur ett-ställning görs av biten D2 i M (D7 ... D0)

- a) CBR M,0b000000100
- b) LDS R20,M  
LDI R21,0x04  
OR R20,R21  
STS M,R20
- c) LDS R20,M  
ANDI R20,0b000000100  
STS M,R20
- d) LDS R20,M  
ORI R20,0x02  
STS M,R20
- e) SBR M,0b000000100

## 18. PORTB

*avr\_io\_b.aw2*

Antag att PORTB har värdet 0x37. Vilket värde har PORTB efter att följande programsats har utförts?

```
IN R20,PORTB
ORI R20,0xC0
OUT PORTB,R20
```

Ange svaret på hexadecimal form.

### 19. PORTB

*avr\_io\_c.aw2*

Antag att PORTB har värdet 0xB3. Vilket värde har PORTB efter att följande har utförts:

```
IN R20,PORTB
ANDI R20,0xC0
OUT PORTB,R20
```

Ange svaret som ett 8 bitars tal.

### 20. BITMASK

*avr\_bit\_mask\_and\_a.aw2*

Vilken bitmask skall användas för att nollställa bitarna D3,D2,D1 och D0 med hjälp av en AND-operation?

Ange talet som ett 8 bitars binärt heltal på formen: xxxxxxxx

## BILAGA A. SVARSTYPER

Olika typer av frågor förekommer såsom multipelt val, numeriskt värde, etc., varje typ av fråga har sin typ av svarsalternativ.

- Ordagrann jämförelse
  - Jämförelse görs bokstav för bokstav.
- Multipelt val (multiple choice)
  - Svarsalternativen är en bokstav A, B, C, etc. Rättningen är okänslig för om svaret anges med VERSALER eller gemener.
- Sant/falskt, rätt/fel, och ja/nej
  - Frågor av typen sant/falskt, rätt/fel och a/nej. anges med svarsvärden för det som är RÄTT/SANT/TRUE/YES med bokstäverna R, S, T, och Y och för det som är FEL/FALSE/NO med bokstäverna F och N. Svarsalternativet fungerar både med VERSALER och gemener. För sant kan även siffran 1 användas och för falskt siffran 0.
- Kommaseparerad lista av ord
- Numeriskt värde med storhet
  - Numeriska svarsvärden är heltal eller reella tal och anges med DECIMALPUNKT istället för decimalkomma.
  - Exempel 1: 125.3
- Numeriskt värde med storhet och enhet
  - Enheter som understöds:
    - \* Alla enheter som representeras med 1 tecken understöds, t.ex.  $s$  för sekunder,  $m$  för meter, etc.
    - \* ohm - anges som ohm.
    - \* Ett blanktecken skall finnas mellan det numeriska värdet och enheten.
  - Prefix som understöds t.ex. n för ns (nanosekunder):
    - \* m - milli
    - \* u - mikro
    - \* n - nano
    - \* p - pico
    - \* k - kilo
    - \* M - Mega
    - \* G - Giga
    - \* T - Terra
- Enkelt matematiskt uttryck
  - Exempel 1:  $y=x^{**2}+3*x+4$
  - Exempel 2:  $24*12+3$
- Hexadecimal sträng
  - Exempel 1: bb45
  - Exempel 2: 0x00A6



## – Exempel 3: 0XFFE3

Svaren är ej känsliga för extra vita-tecken (blanktecken, etc.) då t.ex tre blanktecken efter varann ersätts med ett blanktecken.

**A.1. Decimalpunkt.** I alla svar som kräver ett reellt tal som svar används decimalpunkt istället för decimalkomma.

**A.2. Enkla matematiskt uttryck.** Ett enkelt matematiskt uttryck har en syntax som motsvaras av det man finner i programmeringsspråk som C, MatLab, Ruby, etc. Då små detaljer kan skilja olika programmeringsspråk åt.

Observera att skillnad föreligger mellan HELTALSDIVISION och division av reella all så kallad FLYTTALSDIVISION. Ex 1:  $10 / 6$  ger som resultat 1, däremot så ger  $10.0 / 6$  resultat 1.666667. Den enkla regeln är att om en av operanderna vid division är ett reellt tal fås en flyttalsdivision, dvs en helt normalt resultat med decimaldelar. Om båda operanderna är ett heltal erhålls alltid ett heltalsresultat av division dvs decimaldelarna kastas.

I exemplena som följer i tabellen har variablerna a och b heltalsvärdena 10 respektive 20 och c och d de reella värdena 10.0 respektive 20.0.

Operator	Beskrivning	Exempel
+	Additionsoperatör	a+b ger som resultat 30
-	Subtraktionsoperatör	a-b ger som resultat -10
*	Multiplikationsoperatör	a*b ger som resultat 200
/	Divisionsoperatör	a/b ger som resultat 0 (heltalsdivision) c/d ger som resultat 0.5 (flyttalsdivision, reell division)
%	Modulusoperatör	
**	Exponent	a ** b ger a upphöjt i b dvs 10 upphöjt i 20 b ** 2 ger som resultat 400

Följande matematiska funktioner kan användas:

Funktion	Beskrivning	Exempel
acos	Arcuscos	
asin	Arcussinus	
atan	Arcustangens	
cos	Cosinus	
exp	Exponentialfunktionen	
log	e-logaritmen (ln)	
log10	10-logaritmen	log10(100) ger värdet 2
log2	2-logaritmen	
sin	Sinus	sin(PI/2) ger värdet 1.0
sqrt	Kvadratroten	sqrt(4) ger värdet 2
tan	Tangens	

**A.3. Enkla differentialekvationsuttryck.** En differentialekvation som t. ex.

$$\frac{d^2y}{dt^2} + 3\frac{dy}{dt} + 2y = 3\frac{du}{dt} + 2u$$

anges som ett enkelt differentialekvationsuttryck anges på följande form:

$$y'' + 3y' + 2y = 3u' + 2u$$

Den enkla regeln är att varje derivering anges med en apostrof, dvs för andraderivatan blir det två apostrofer efter varann.