

EDA215 / EDA432 / INN790

Digital- och dator teknik för Z / It / GU

**Gamla och
nya kursen**

Tentamen

Torsdag 23 augusti 2007, kl. 14.00 - 18.00 i M-salar

Examinatorer

Rolf Snedsböl, tel. 772 1665

Kontaktpersoner under tentamen

Som ovan.

Tillåtna hjälpmedel

Häftet

Instruktionslista för FLEX

och

Instruktionslista för MC6809

eller

Instruktionslista för CPU12

I dessa får rättelser och understrykningar vara införda, inget annat.

Tabellverk och miniräknare får ej användas!

Allmänt

Siffror inom parentes anger full poäng på uppgiften. **Full poäng kan fås om:**

- redovisningen av svar och lösningar är läslig och tydlig. **OBS!** Ett lösningsblad får endast innehålla redovisningsdelar som hör ihop med en uppgift.

- din lösning ej är onödigt komplicerad.
- du motiverat dina val och ställningstaganden
- redovisningen av en hårdvarukonstruktion innehåller funktionsbeskrivning, lösning och realisering.
- redovisningen av en mjukvarukonstruktion i assembler är fullständigt dokumenterad, d v s är redovisad både i strukturform (flödesplan eller pseudopråk) och med kommenterat program i assemblerspråk, om inget annat anges i uppgiften.

Betygsättning

För godkänt slutbetyg på kursen fordras att både tentamen och laborationer är godkända. Tentamen ger slutbetyget:

$20p \leq \text{betyg } 3 < 30p \leq \text{betyg } 4 < 40p \leq \text{betyg } 5$

Lösningar

anslås på kursens www hemsida (EDA215).

Betygslistan

anslås såsom anges på kursens hemsida (EDA215).

Granskning

Tid och plats anges på kursens hemsida (EDA215).

© Stig-Göran Larsson och Rolf Snedsböl, 2007

Avdelningen för dator teknik

**Institutionen data- och informationsteknik
CHALMERS TEKNISKA HÖGSKOLA**



1. Koder, talomvandling, aritmetik och flaggor

I uppgift a-d nedan används 5-bitars tal. $X=01001$ och $Y=01100$

- a. Visa med penna och papper hur räkneoperationen $R = X - Y$ utförs i en dator (i en ALU). **(1p)**
- b. Ange sedan flaggbitarna N, Z, V, C **(1p)**
- c. Tolka bitmönstren R, X och Y som tal *utan* tecken och ange dess decimala motsvarighet **(1p)**
- d. Tolka bitmönstren R, X och Y som tal *med* tecken och ange dess decimala motsvarighet. **(1p)**
- e. Studera bitmönstren $(11011100)_2$ och $(01011001)_2$.

Kan bitmönstren representera följande:

(Ge ditt svar i tabellform enligt:)

- 1) decimaltal utan tecken
- 2) ett positivt tvåkomplementstal
- 3) tecken belopps tal
- 4) NBCD-tal
- 5) Gray-kod
- 6) Ett 7-bitars ord utökat med en jämn paritetsbit

	11011100	01011001
1	Ja/Nej	Ja/Nej
2		
osv...		

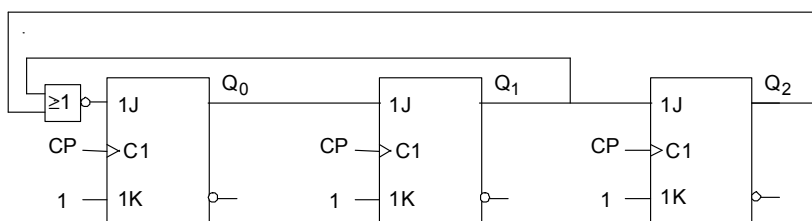
(3p)

2. Digitalteknik, kombinatoriska nät

- a. Man behöver en INVERTERARE men har bara 2- ingångs XOR-grindar. Kan man använda XOR-grinden för detta ändamål och i så fall hur kopplar man upp INVERTERAREN med XOR-grinden? **(1p)**
- b. Den booleska funktionen $f(x,y,z) = xy + xz + y'z$ är given. Skriv $f(x,y,z)$ på konjunktiv normal form och på konjunktiv minimal form. **(2p)**
- c. En krets med fyra binära insignaler x, y, z, w och en utsignal f_a skall konstrueras. Insignalerna utgör siffror i det binära talet $(xyzw)_2$. Utsignalen $f_a = 1$ om $(xyzw)_2 \leq 6$ och $f_a = 0$ om $(xyzw)_2 > 6$. 2-ingångars NAND-grindar och INVERTERARE får användas. **(6p)**

3. Digitalteknik, sekvensnät

- a. *Analys.* En räknare är given nedan. Sätt upp en tabell med ”detta tillstånd”, ”vippornas insignaler JK” och ”nästa tillstånd”. Ange även räknarens utsekvens i en tillståndsgraf. **(6p)**



(6p)

- b. Visa hur du kan bilda en JK-vippa med hjälp av en SR-vippa och två AND-grindar. **(2p)**

4. Styrenheten för FLEX

- a. I tabellen nedan visas RTN-beskrivningen för EXECUTE-sekvensen för en av FLEX-processorns instruktioner. NF i tabellens sista rad anger att nästa tillstånd (state) skall vara det första i FETCH-sekvensen. Rita en tabell där du anger State nr (0..3) och Styr signaler. Endast styr signaler = 1 skall anges.

Du kan utelämna RTN-beskrivningen i din tabell.

(1p)

State nr	RTN-beskrivning	Styr signaler (=1)
0	PC→MA, PC+1→PC	
1	M→MA	
2	M-1 →R, Flaggor→CC	
3	R→M, NF	

- b. Förklara med ord vad instruktionen ovan utför i varje klockcykel. Skriv instruktionen med assemblerspråk. (2p)
- c. Rita en tabell motsvarande den ovan, som visar utförandefasen för maskininstruktionen **JSR Adr** för FLEX-processorn. I instruktionslistan för FLEX-processorn beskrivs instruktionen enligt följande tabell. (6p)

Instruktion		Adressering			Operations- beskrivning*	Flaggor			
Operation	Beteckning	Absolute				3	2	1	0
		OP	#	~	N	Z	V	C	
Jump to subroutine	JSR Adr	69	2	7	S-1 → S PC → M(S) Adr → PC	•	•	•	•

5. Småfrågor och assemblerprogrammering för FLEX

- a. Ett 24-bitars tal är lagrad på adresserna 05, 06 och 07 med mest signifikanta byten på adress 05. Skriv en instruktionssekvens som negerar (två-komplementerar) detta 24-bitars tal (3p)
- b. Nedan visas ett assemblerprogram. Ange maskinkoden för programmet. Det skall klart framgå hur du beräknat instruktionens offset

Assemblerprogram:

```

ORG $30
Start LDX #$0C
LOOP LDA ,X+
      STA $FE
      DECB
      BPL LOOP
      LDA 12
      NOP

```

(4p)

6. Assemblerprogrammering för MC12.**Lös antingen uppgift 6 eller 7, inte båda**

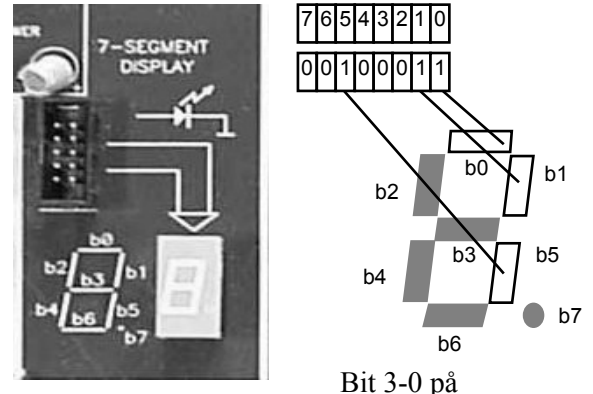
Om du är registrerad på gamla kursen och haft MC6809-processorn löser du uppgift 7.

Om du är registrerad på nya kursen med MC12 löser du uppgift 6

I labbet använde du 7-sifferindikatorn som var anslutet till utporten på MC12. Vidare använde du som inport 8 strömbrytare (Dip Switch Input).

Du skall skriva ett program som hela tiden läser inporten (strömbrytarna) och skriver värden till utporten (sifferindikatorn).

När bit 7 på inporten är ettställd skall sifferindikatorn släckas helt. När bit 7 på inporten är nollställd skall sifferindikatorn tändas enligt följande beskrivning:



inporten anger vad som skall visas på sifferindikatorn. Om indata är i intervallet [0,9] skall motsvarande decimala siffra visas på sifferindikatorn. Om indata är i intervallet [A,F] skall ett E (Error) visas på sifferindikatorn.

Bitarna 6-4 på inporten kan anta vilka värden som helst.

Skriv detta program!

Du har tillgång till en tabell i minnet med segmentkoder (mönster för sifferindikatorn) enligt **SegCodes FCB \$77, \$22, \$5B, \$6B, etc.**

Tabellen innehåller segmentkoder för siffrorna [0,9].

På adressen "SegCodes" i minnet finns segmentkoden för 0,
 på adressen "SegCodes+1" i minnet finns segmentkoden för 1,
 på adressen "SegCodes+2" i minnet finns segmentkoden för 2,
 etc

(10p)

7. Avbrott och assemblerprogrammering med MC6809**Lös antingen uppgift 6 eller 7, inte båda**

Om du är registrerad på nya kursen med MC12 löser du uppgift 6

Om du är registrerad på gamla kursen och haft MC6809-processorn löser du uppgift 7.

Redogör för hur avbrott går till i ett MC6809-system, där du har en (endast en) yttre enhet ansluten till IRQ-ingången på MC6809:an!

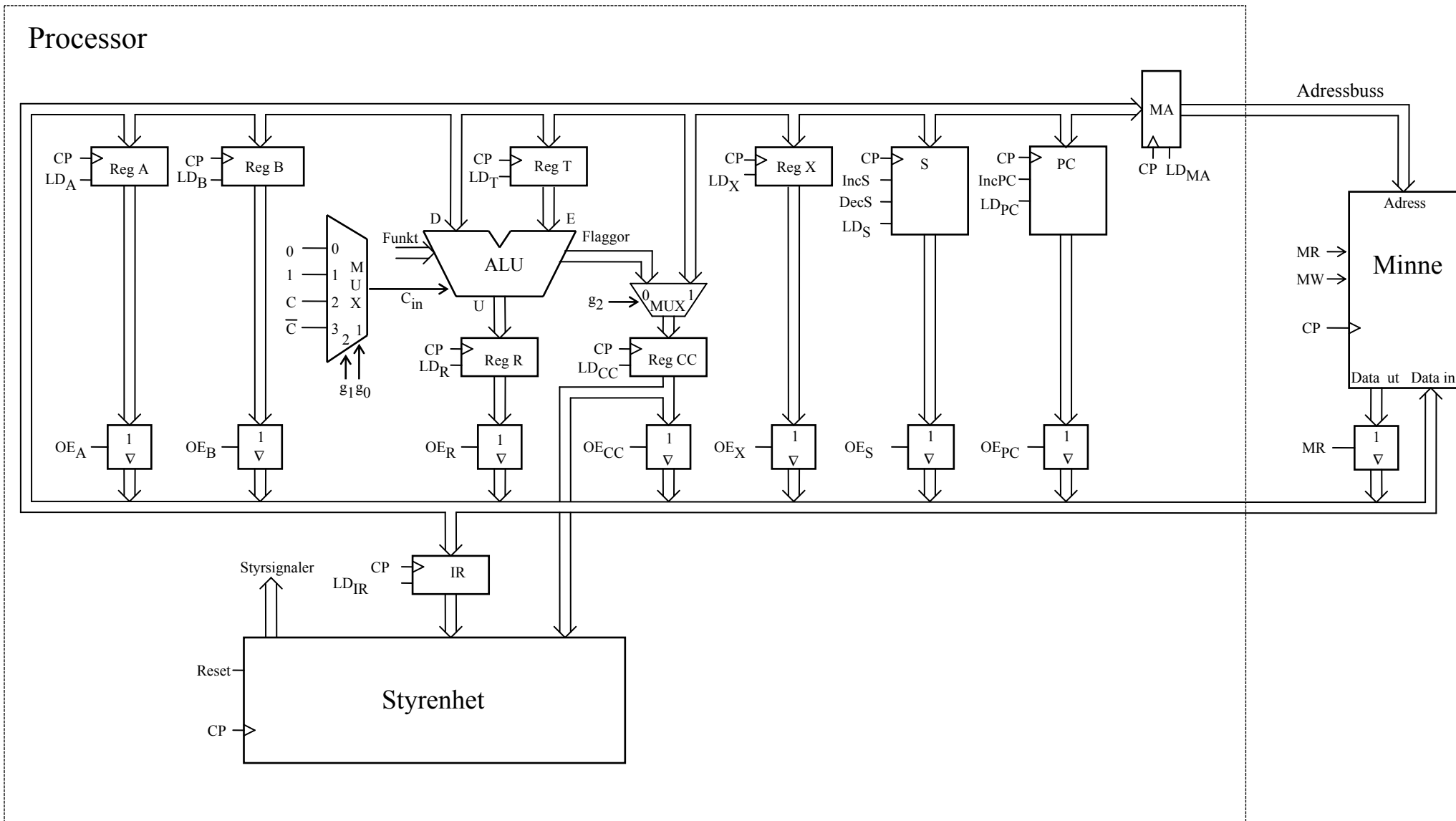
Ditt svar skall bland annat innehålla

- en övergripande beskrivning av vad avbrott är och hur det fungerar
- en beskrivning av eventuell yttre hårdvara (en skiss kan vara mycket beskrivande)
- en beskrivning av de olika programrutiner som är förknippade med avbrott (eventuellt pseudokod / assembler)

(10)

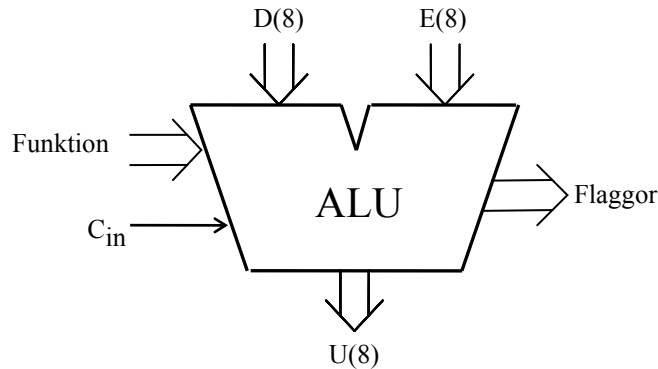
FLEX-datorn

Bilaga 1



ALU:ns funktion

Bilaga 2



ALU:ns operation (logik- eller aritmetik-) på indata **D**, **E** och **C_{in}** bestäms av insignalerna **Funktion** [**F** = (**f₃**, **f₂**, **f₁**, **f₀**)] enligt tabellen nedan. I kolumnen Operation förklaras, när det behövs, hur operationen utförs. Med "+" och "-" avses **aritmetiska operationer**.

f ₃ f ₂ f ₁ f ₀	U = f(D,E,C _{in})	
	Operation	Resultat
0 0 0 0	bitvis nollställning	0
0 0 0 1		D
0 0 1 0		E
0 0 1 1	bitvis invertering	D _{1k}
0 1 0 0	bitvis invertering	E _{1k}
0 1 0 1	bitvis OR	D OR E
0 1 1 0	bitvis AND	D AND E
0 1 1 1	bitvis XOR	D XOR E
1 0 0 0	D + 0 + C _{in}	D + C _{in}
1 0 0 1	D + FFH + C _{in}	D - 1 + C _{in}
1 0 1 0		D + E + C _{in}
1 0 1 1	D + D + C _{in}	2D + C _{in}
1 1 0 0	D + E _{1k} + C _{in}	D - E - 1 + C _{in}
1 1 0 1		0
1 1 1 0		0
1 1 1 1	bitvis ettställning	FFH

Flaggorna är utsignaler och för de gäller:

Carryflaggan (C) är minnessiffran ut (carry-out) från den mest signifikanta bitpositionen (längst till vänster) när en aritmetisk operation utförs av ALU:n.

Vid **subtraktion** gäller för denna ALU att **C = 1 om lånesiffra (borrow) uppstår och C = 0 om lånesiffra inte uppstår**.

Carryflaggans värde är 0 vid andra operationer än aritmetiska.

Overflowflaggan (V) visar när en aritmetisk operation ger "overflow" enligt reglerna för 2-komplementaritmetik.

V-flaggans värde är 0 vid andra operationer än aritmetiska.

Zeroflaggan (Z) visar när en ALU-operation ger värdet noll som resultat på U-utgången.

Signflaggan (N) är identisk med den mest signifikanta biten (teckenbiten) av utsignalen U från ALU:n.

Half-carryflaggan (H) är minnessiffran (carry) mellan de fyra minst signifikanta och de fyra mest signifikanta bitarna i ALU:n.

H-flaggans värde är 0 vid andra operationer än aritmetiska.

Tentamen i Digital och Datorteknik, 2007-08-23

- 1a) $R=X-Y$ utförs som $R=X+Y_{1komp}+1$ $Y_{1komp} = 10011$.
- 1b) $N=1; Z=0; V=0; C_5=0 \Rightarrow C=1$
- 1c) $X=9; Y=12; R=29$ ($9-12 \neq 29$, verkar rimligt ty $C=1$)
- 1d) $X=9; Y=12; R=-3$ ($9-12=-3$, verkar rimligt ty $V=0$)
- 1e) JJ; NJ; JJ;NJ; JJ; NJ

	00011
X	01001
+Y _{1komp}	+ 10011
+1	+ 1
=R	= 11101

Uppg 2

2a) JA, XOR-grinden kan användas om ena ingången ansluts till en ETTA.

2b)

x	y	z	xy	xz	y'z	f
0	0	0	0	0	0	0
0	0	1	0	0	1	1
0	1	0	0	0	0	0
0	1	1	0	0	0	0
1	0	0	0	0	0	0
1	0	1	0	1	1	1
1	1	0	1	0	0	1
1	1	1	1	1	0	1

		yz			
		00	01	11	10
x	0	0	1	0	0
	1	0	1	1	1

Minimal form: $f(xyz) = (y+z)(x+y')$

Normal form: $f(xyz) = (x+y+z)(x+y'+z)(x+y'+z')(x'+y+z)$

2c)

Analys / Funktionsbeskrivning

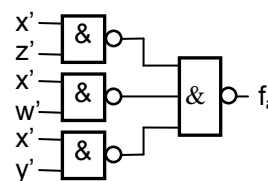
x	y	z	w	f _a
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Lösning o minimering

		zw			
	f _a	00	01	11	10
xy	00	1	1	1	1
	01	1	1	0	1
	11	0	0	0	0
	10	0	0	0	0

$f_a = x'y' + x'z' + x'w'$

Realisering



Upp 3

- 3a)** Se lösning uppgift 5.12 i blåa boken del 1
3b) Se sid 5.15, Figur 5.26 i kursboken

Upp 4**4a)**

State nr	RTN-beskrivning	Styrsignaler (=1)
0	PC→MA, PC+1→PC	OE _{PC} , LD _{MA} , IncPC
1	M→MA	MR, LD _{MA}
2	M-1 →R, Flaggor→CC	MR, f ₃ , f ₀ , LD _R , LD _{CC}
3	R→M, NF	OE _R , MW, NF

4b)

- 0) Förbered för läsning av adressoperand i minnet, Öka PC med ett
- 1) Läs adressoperanden från minnet till register MA för att förbereda läsning av data
- 2) Läs data från minnet, minska med ett ospara resultatet i register R. Påverka flaggbitarna.
- 3) Skriv resultatet till minnet

Instruktionen är DEC Adr

4c)

State nr	RTN-beskrivning	Styrsignaler (=1)
0	S-1→S, PC→MA, PC+1→PC	DecS, OE _{PC} , LD _{MA} , IncPC,
1	M→T	MR, LD _T
2	S→MA	OE _S , LD _{MA}
3	PC→M, T→R	OE _{PC} , MW, f ₁ , LD _R
4	R→PC	OE _R , LD _{PC} , NF

Upp 5**5a)** (Negera = Invertera och addera ett)

```

COM    $05    Invertera
COM    $06    Invertera
COM    $07    Invertera
INC    $07    Addera ett till låga delen
BCC    Klart  Någon Carry
INC    $06    . isåfall addera
BCC    Klart  .. och en gång till
INC    $05
Klart  ---

```

5b) Endast Hexkoder: 11, 0C, 7B, 13, FE, 45, 5C, FA, 0B, 0C, 00

Upp 6

OCH en flödesplan!

	LDX	#SegCodes	Pekare
Start	LDAA	Inport	Läs strömbrytare
	BPL	B7Zero	Om Bit 7=0
	CLR	Outport	Släck
	BRA	Start	
B7Zero	ANDA	#\$0F	Maska fram b3-b0
	CMPA	#9	[0,9]?
	BHI	Error	
	LDAB	A,X	Hämta segmentkod för [0,9]
	STAB	Outpoprt	Visa siffra
	BRA	Start	
Error	LDAB	;%01011101	Kod för E
	STAB	Outport	Visa siffra
	BRA	Start	