

# DATORTEKNIK MAJ-10

1. a) 1 1000 0100 000 1101 0000 0000 0000 0000

S=1 NEGATIVT

EP  $E = 1000\ 0100_2 = 132 \Rightarrow 132 - 127 = 5$

M = 000 1101 0000 0000 0000 0000  $\Rightarrow 1.000\ 1101$

$-1.000\ 1101 \cdot 2^5 = -100011.01 = -35,25$

b)  $17,625 = 10001101 = 1.0001101 \cdot 2^4$

POSITIVT S=0

EP  $127 + 4 = 131 = 1000\ 0011 = E$

M = 000 1101 0000 0000 0000 0000

0 1000 0011 000 1101 0000 0000 0000

c)  $+50 = \underline{00110010}_{2c}$   $\begin{matrix} 11001101 & + & \text{kompl} \\ + & & 1 \end{matrix}$   
 $-50 = \underline{11001110}_{2c}$   $\begin{matrix} 11001110 \end{matrix}$

1	2	3	4	5	6	7	8	9	10
D	F	H	I A	G	K	L	A	E A	C

3. 1a) PARALLELL ADC (FLASH) 5448

- 1b) FÖRDELAR: MYCKET SNABB OMRÖMNING
- 1c) NACKDELAR: ANVÄNDER KOMPARATÖRER

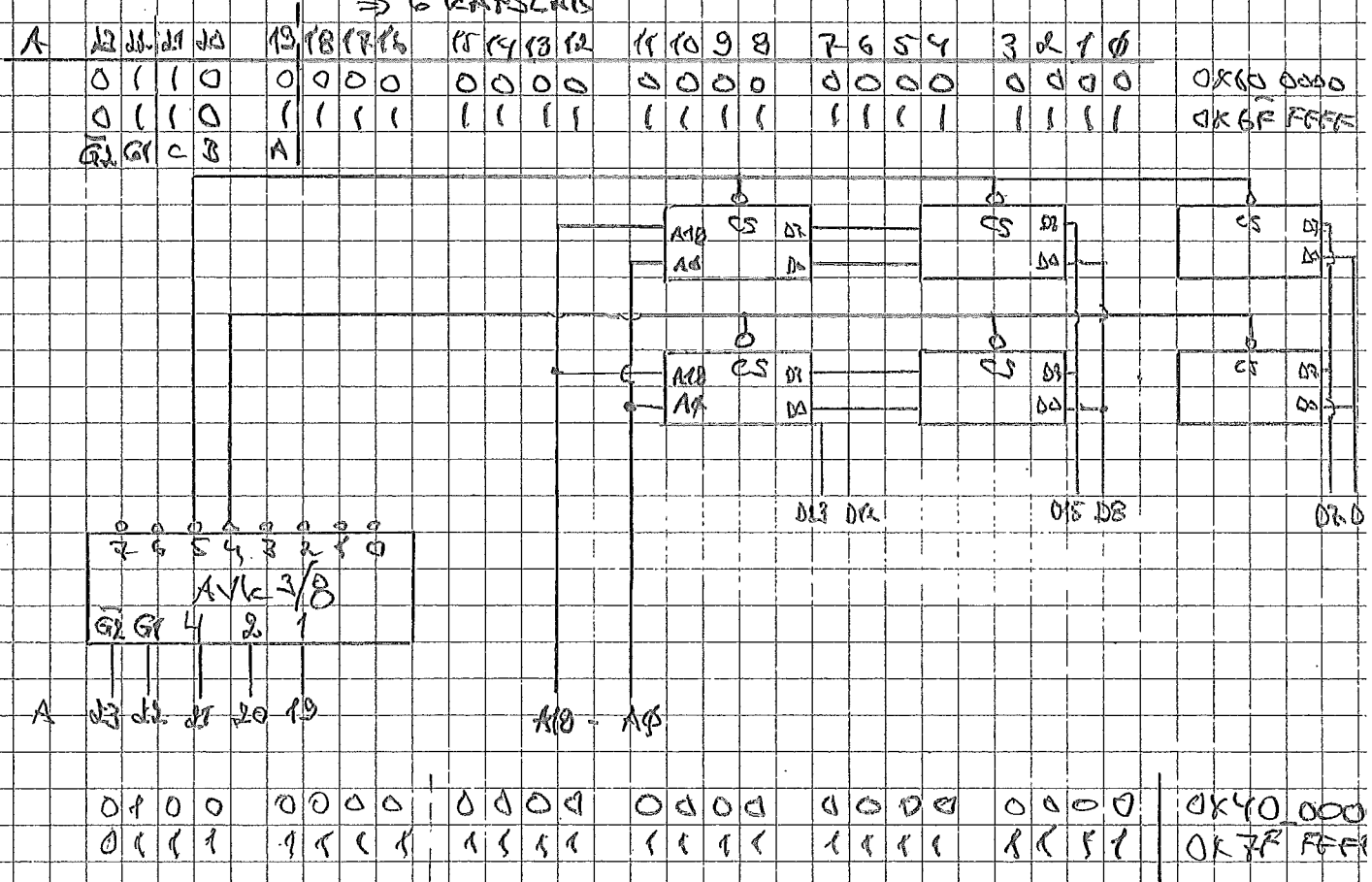
FUNKTION: UIN JÄMFÖR MED ANV. REFERENS NIVÅN GEMEDELT  
 I ETT PAR AV PRECISIONSBESTÄNDER (SA. DELAR

JÄMFÖRLESEN GÖR I KOMPARATÖREN (SNABB OMRÖMNING)  
 $V_+ > V_- \Rightarrow HÖG \Rightarrow$  "TERTIÖMETERKOD"  
 $V_+ < V_- \Rightarrow LÅG$

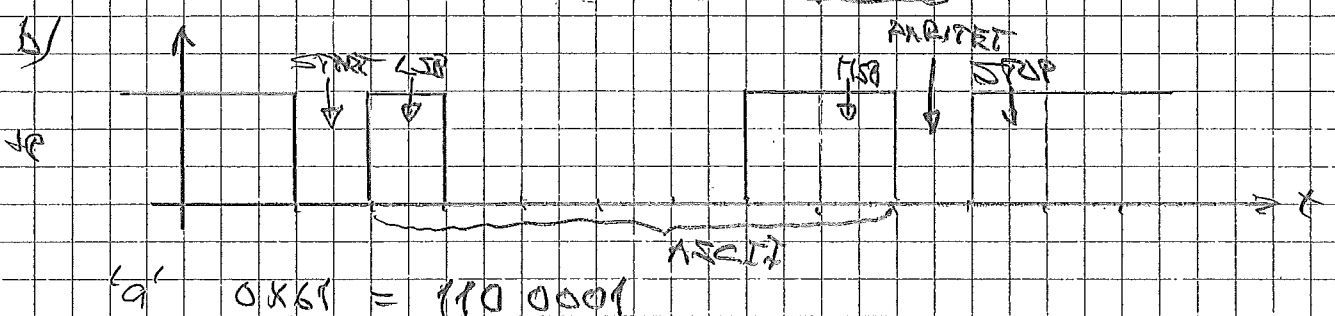
"TERTIÖMETERKODEN" OMRÖMNAS TILL BINÄRS

4.  $2^{19} = 0,5 M = 512k \times 8$  / KAPSEL

4p TOTAL: 1M x 24 BITAR 1 ADDR. DIR. 0x600000 - 0x6FFFF  
 $\Rightarrow 6$  KAPSLAR



5. a) 1 SPARK + 2 DATA + 1 PARITET + 1 STOP = 10 BITAR  
 19200 BIT/S  $\Rightarrow$  1920 TKB/SEK



SP	1	2	3	4	5
	S	F	S	S	F

0x60

8. a) 41 BYTE 10 INSTR  $\Rightarrow$  40 BYTE + VISA 1 BYTE

8p b) VISA FORMAS PA ADDR. 0x1008

6

C:\Datorteknik\_tentauppg\Rinnandeljus\_maj-10\lab1\_1.s

;Rinnande ljus maj-10

;100525

#include "s3c3410x\_asm.h"

```

CODE32
ORG 0X00
B INIT

INIT ORG 0X1000
LDR SP,=0X2000
LDR R0, =rPCON0
LDR R1, =0X2BF
STRh R1, [R0] ;Port0 utport

; -----
MAIN LDR R0, =rPDAT0
MOV R1, #0
STRb R1, [R0] ;0-ställ Port0
BL DELAY
MOV R2, #8 ;räknare
MOV R3, #0X80
LOOP1 EOR R1, R1, R3 ;invertera den bit som pekas ut av R3 JA
STRb R1, [R0] ;skicka ut på Port0
BL DELAY
MOV R3, R3, LSR #1 ;nästa bit (högerskifta en etta)
SUBS R2, R2, #1 ;håll reda på antalet skift
BEQ LOOP2
B LOOP1

LOOP2 MOV R2, #7
MOV R3, #0X80
LOOP3 EOR R1, R1, R3
STRb R1, [R0]
BL DELAY
MOV R3, R3, LSR #1
SUBS R2, R2, #1
BNE LOOP3
B MAIN

DELAY MOV PC, LR

; -----

END

INIT ORG 0X1000
LDR SP,=0X2000
LDR R0,=rPCON0
LDR R1,=0X2BF
STRh R1, [R0]

; -----
MAIN LDR R0,=rPDAT0
LDR R1,=0XFF00FF00
LOOP STRb R1, [R0]
BL DELAY
MOV R1,R1,ROR #1
B LOOP

DELAY MOV PC,LR

END

```

ALT:

9.	HANDLER	STARTD	SP, {R0-R4, LR}	1P
4P		LDR	R0, =INTEND	
		LDR	R1, [R0]	
	R0	ANDS	R1, R1, #0x01	
		BEQ	T8	1P
		BL	ALARM	
		BIC	R1, R1, #0x01	1P
		R	EXIT	
	T8	ANDS	R2, R1, #0x100	
		BEQ	EXIT	(6P)
		BL	TIMER	
		BIC	R1, R1, #0x100	(6P)
	EXIT	STR	R1, [R0]	
		LDRD	SP, {R0-R4, LR}	(6P)
		SUBS	PC, LR, #4	1P

10

C:\Datorteknik\_tentauppg\Analys2\_maj-10\lab1\_1.s

```
#include "s3c3410x_asm.h"
```

```
; ----- Konstanter -----  
  
RAM_START      EQU      0X00001000  
  
; ----- Initieringskod -----  
                CODE32  
  
                ORG      0X0  
                B        Rad3  
  
Rad2:           ORG      RAM_START  
Rad3:           LDR      R2, =TEXT      ;=saknas  
                LDR      R3, =SVAR     ;adressen till SVAR till R3  
Rad4:           MOV      R1, #10  
                MOV      R4, #0        ;0-ställ resultatregister  
  
LOOP  
Rad6:           CMP      R1, #0  
Rad7:           BEQ      FINISH  
                SUB      R1, R1, #1    ;minskning flyttad från rad5  
Rad8:           LDRb     R0, [R2]      ;läs av tecken ur TEXT (b-8 bitar)  
                SUB      R0, R0, #48   ;måste minska med ASCII '0'  
Rad9:           ADD      R4, R4, R0  
Rad10:          ADD      R2, R2, #1  
Rad11:          B        LOOP  
  
FINISH          STRh     R4, [R3]  
QUIT           B        QUIT  
  
TEXT           DC8      "13128571983758134612846182746"  
SVAR           DS16     1
```

END

(11)

;a) Subrutin BITSET som 1-ställer en angiven bit  
;b) Subrutin RZERO som undersöker en operand.Index för minst sign.  
; 0-nollan utparameter  
;c) Subrutin LSB\_SET som 1-ställer minst sign. 0-an i en minnescell.  
; Index för den nolla som 1-ställts utparameter

```
#include "s3c3410x_asm.h"
```

```
                ORG     0X1000

; Initiera stackpekare
MAIN    LDR     SP,=0X2000
; -----Huvudprogram-----
AUPPG   LDR     R4,=0X000FFFF
        MOV     R5,#17
        BL     BITSET
BUPPG   LDR     R4,TAL
        BL     RZERO
CUPPG   LDR     R4,=TAL
        BL     LSB_SET
STOP    B      STOP

; -----Subrutin BITSET-----
BITSET  STMFD   SP!, {R8,LR}
        MOV     R8,#0X01           ;1-ställ bit0
        MOV     R8,R8,LSL R5      ;skifta 1:an till rätt position
        ORR     R2,R4,R8          ;1-ställ utpekad bit
        LDMFD   SP!, {R8,PC}

; -----Subrutin RZERO-----
RZERO   STMFD   SP!, {R8,R9,LR}
        LDR     R2,=0XFF          ;0xFF om ingen 0:a
        CMP     R4,#0xFFFFFFFF    ;kolla om ingen 0:a
        BEQ     KLART
        MOV     R8,#1             ;bestäm minst sign. 0:a i R4
        MOV     R2,#0
LOOP    AND     R9,R4,R8
        CMP     R9,#0
        BEQ     KLART
        ADD     R2,R2,#1          ;updatera index
        MOV     R8,R8,LSL #1      ;nästa position
        B      LOOP
KLART   LDMFD   SP!, {R8,R9,PC}

; -----Subrutin LSB_CLR-----
LSB_SET STMFD   SP!, {R4,R5,LR}
        LDR     R4,[R4]           ;(R4)<--((R4)), läs in tal
        BL     RZERO              ;bestäm indexvärde för LS 0:a
        CMP     R2,#0XFF
        BEQ     BRYT              ;BLT
        MOV     R5,R2             ;(R5)<--(R2)
        BL     BITSET            ;1-ställ
        MOV     R4,R2             ;(R4)<--(R2)
        MOV     R2,R5             ;index tillbaks i R2
BRYT    MOV     R8,R4             ;mellanlagra det modif. värdet i R8
        LDMFD   SP!, {R4,R5,LR}  ;återställ
        STR     R8,[R4]           ;skriv tillbaka det modif. värdet
        MOV     PC,LR

TAL     DC32    0xFFFFFFFFE

END
```

12

```

;Pelle behöver en sorteringsmaskin för LEGO-bitar. Du ska i den här
;uppgiften programmera den ingående styrenheten, som är baserad på ARM-processorn.
;LEGO-bitarna kommer på ett transportband, som passerar förbi tre
;fotoceller P1, P2 och P3 placerade med 5 cm mellanrum (se fig.).
;Tre olika typer av LEGO-bitar förekommer: Små bitar med längden 4 cm,
;mellanstora bitar med längden 8 cm samt stora med längden 12 cm.
;De små bitarna ska knuffas av bandet och ner i en låda, så fort de har
;passerat P2. Detta sker genom att styrsignalen KNUFF1 aktiveras.
;De mellanstora bitarna ska på samma sätt knuffas av bandet, när de har
;passerat P3, (styrsignal KNUFF2). De stora bitarna ska fortsätta framåt på bandet.
;Avståndet mellan LEGO-bitarna är alltid minst 10 cm.
;Insignaler till styrenheten: P1 (P0:2), P2 (P0:1) och P3 (P0:0) (aktivt höga).
;Utsignaler från styrenheten: KNUFF1 (P0:7) och KNUFF2 (P0:6) (aktivt höga)
;Uppgift: Skriv ett väl kommenterat ARM-program för styrningen.
;
; -----

```

```

#include "s3c3410x_asm.h"

```

```

CODE32
ORG 0X0
B INIT

ORG 0X1000

```

```

; Initiera stackpekare
INIT LDR SP,=0X2000
LDR R0,=rPCON0
LDR R1,=0X2B8 ;P0:7-4 ut, P0:2-0in
STRh R1,[R0]

```

```

; -----
MAIN MOV R1,#0
LDR R0,=rPDAT0
STRb R1,[R0] ;0-ställ Port0

```

```

START LDRb R1,[R0]
ANDS R2,R1,#0X04 ;P1?
BEQ START

```

```

WAIT1 LDRb R1,[R0]
ANDS R2,R1,#0X02 ;P2?
BEQ WAIT1
LDRb R1,[R0]
ANDS R2,R1,#0X04 ;P1?
BEQ LITEN

```

```

WAIT2 LDRb R1,[R0]
ANDS R2,R1,#0X01 ;P3?
BEQ WAIT2
BNE MELLAN

```

```

WAIT3 LDRb R1,[R0]
ANDS R2,R1,#0X04 ;P1?
BNE WAIT3
B MAIN

```

```

LITEN LDRb R1,[R0]
ANDS R2,R1,#0X02 ;vänta tills P2 passerats
BNE LITEN
ORR R1,R1,#0X80 ;KNUFF1
STRb R1,[R0]
B MAIN

```

```

MELLAN LDRb R1,[R0]
ANDS R2,R1,#0X01 ;vänta tills P3 passerats
BNE MELLAN
ORR R1,R1,#0X40 ;KNUFF2
STRb R1,[R0]
B MAIN

```