

2. SUBROUTINEN - ANROPAS MED BL
 - (PC) SPARAS I LÄNKREGISTERET
 - PC LADDAS MED SUBROUTINENS STARTADRESS
- AVBROTT ANROPAS AV EXTERNA SIGNALER
 PC O CPSR SPARAS UMBÄN

3. sp
 0 1000 0000 101 0000 0000 0000 0000 0000
 S E M

S=0 ⇒ POSITIVT

E=1000 0000 ⇒ 128 ⇒ E-127=1 ∴ 2^1

+ 1.101 · 2^1 = +11.01₂ = 3.25

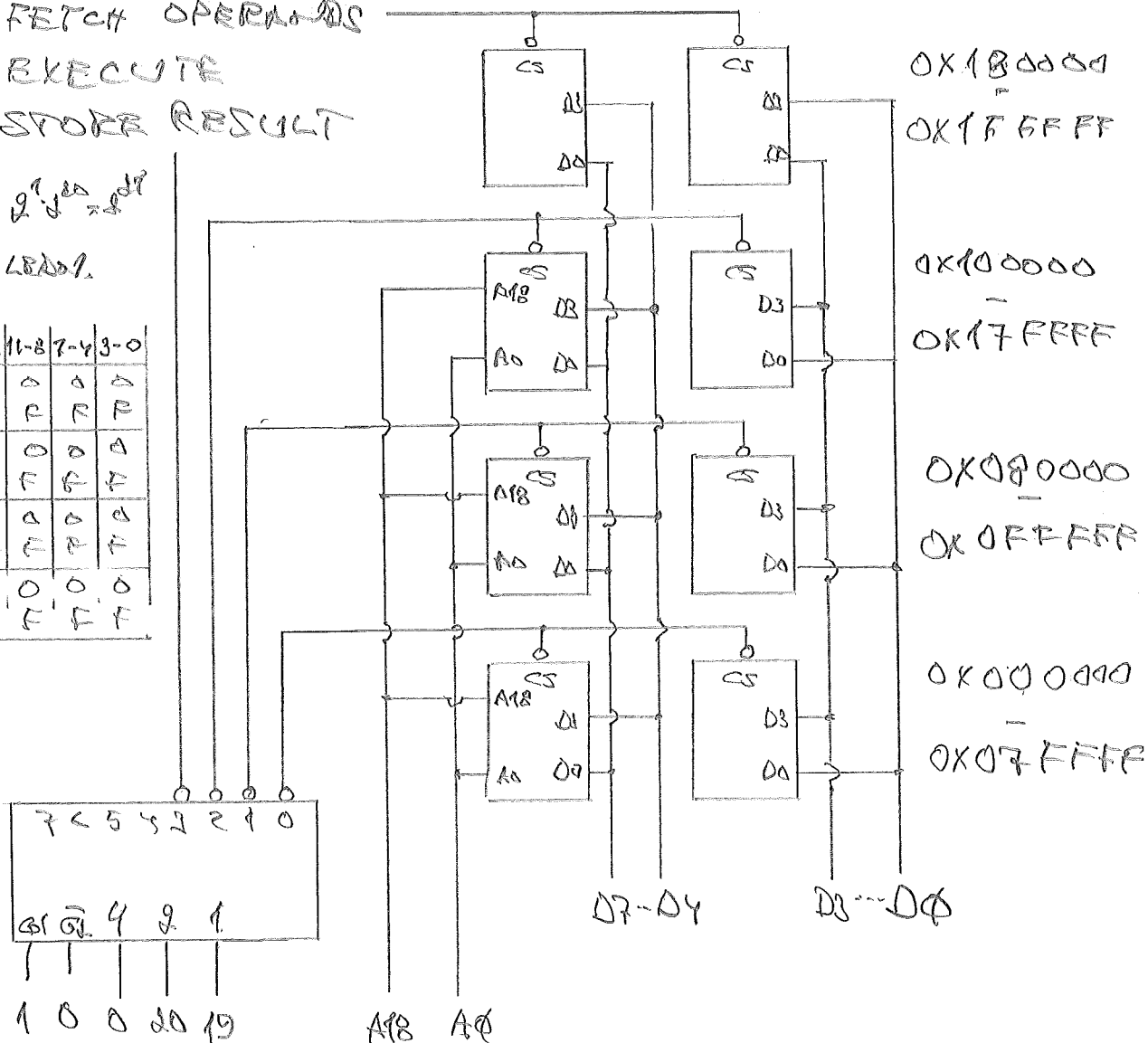
4. FETCH
DECODE
 EVALUATE ADDRESS
 FETCH OPERANDS
 EXECUTE
 STORE RESULT

FA: 19-20

ÄVN. 8

5. $2^7 \Rightarrow 2^1 \cdot 2^6 = 2^7$
 2⁷ ADDR. LEAD.

A	20	19	18	15-12	11-8	7-4	3-0
0	0	000	1	0	0	0	0
D	0	111	F	F	F	F	F
0	1	000	0	0	0	0	0
D	1	111	F	F	F	F	F
1	0	000	0	0	0	0	0
1	0	111	F	F	F	F	F
1	1	000	0	0	0	0	0
1	1	111	F	F	F	F	F



10

```

ORG      0X1000
MAIN     LDR      SP,=0X2000
         LDR      R1,=0X55555555
         LDR      R2,=0XAAAAAAAA

;a)      ANDS     R3,R1,R2           R3 = 0X00      Z=1
;b)      ORRS     R3,R1,R2           R3 = 0XFFFFFF N=1
;c)      LDR      R2,=0XAAAAAAB      R3 = 0X00      Z=1, C=1
         ADDS     R3,R1,R2
;d)      LDR      R2,=0XBCBCBCBC     R3 = 0X98989899
         SUBS     R3,R1,R2           N=1, V=1

STOP     B        STOP

```

```

; -----
END

```

11

```

ORG      0X1000
MAIN     LDR      SP,=0X2000
         BL       TABOMV
         B        MAIN

; -----
TABOMV   STMFD   SP!,{R0-R3,LR}      ;spara undan rgistervärden
         LDR      R0,=TABELL1        ;peka ut TABELL1 (källtabell)
         LDR      R1,=TABELL2        ;peka ut TABELL2 (destinationstabell)
         MOV      R2,#10             ;antal tabellelement
LOOP     LDRb    R3,[R0],#1          ;ladda in tabellelement
         ADD      R3,R3,#1           ;inkrementera
         STRb    R3,[R1],#1          ;spara i destinationstabell
         SUBS    R2,R2,#1            ;håll koll på antal tabellelement
         BNE     LOOP
         LDMFD   SP!,{R0-R3,PC}     ;återställ registervärden

; -----
ORG      0X1100
TABELL1  DC8    0X43, 0X21, 0X56, 0X12, 0X54, 0X43, 0X59, 0X85, 0X06, 0X01

ORG      0X1200
TABELL2  DS8    10

```

12

```

ORG      0x00000018
B        IRQHandler

; Avbrottsmaskregister
LDR      R0,=rINTMSK                ; Slå på EINT0 i maskregister
LDR      R1,=0x00000001
STR      R1,[R0]

LDR      R0,=rEINTMOD
LDR      R1,=0x00000001              ; positiv flank
STRh     R1,[R0]

LDR      R0,=rEINTCON ; slå på EINT0 externIntReg
LDR      R1,=1
STRh     R1,[R0]

```

13

;En hiss i ett trevåningshus skall styras med en ARM.
;In- och utsignaler är följande:
;Insignaler: P0:7 ÅKTILL3, P0:6 ÅKTILL2, P0:5 ÅKTILL1, P0:4 ÄRPÅ3,
;P0:5 ÄRPÅ2, P0:4 ÄRPÅ1.
;Utsignaler: P0:1 HISSUPP, P0:0 HISSNED

```
INIT    ORG      0X1000
        LDR      SP,=0X2000      ;(SP)<-- 0x2000
        LDR      R0, =rPCON0
        LDR      R1, =0X3
        STRh     R1, [R0]        ;Port0 P0:7-5 in, P0:1-0 ut
;-----
MAIN    MOV      R1, #0
        LDR      R0, =rPDAT0
        STRb     R1, [R0]        ;0-ställ Port0 (stanna hiss)

        LDRb     R1, [R0]        ;läs in Port0
        ANDS     R2, R1, #0X20    ;känn av Ä1
        BNE     T1
        ANDS     R2, R1, #0X80    ;känn av Ä3
        BNE     T3
        ANDS     R2, R1, #0X40    ;känn av Ä2
        BNE     T2
        B       MAIN

T1      ORR      R2, R2, #0X01
        STRb     R2, [R0]        ;hiss ned
LOOP1   LDRb     R1, [R0]        ;läs in Port0
        ANDS     R2, R1, #0X04    ;känn av Ä1
        BEQ     LOOP1
        B       MAIN

T3      ORR      R2, R2, #0X02    ;hiss upp
        STRb     R2, [R0]
LOOP3   LDRb     R1, [R0]
        ANDS     R2, R1, #0X10    ;känn av Ä3
        BEQ     LOOP3
        B       MAIN

T2      LDRb     R1, [R0]        ;känn av Ä1
        ANDS     R2, R1, #0X04
        BEQ     N2
        B       U2

N2      ORR      R2, R2, #0X01    ;hiss ned
        STRb     R2, [R0]

LOOPN   LDRb     R1, [R0]
        ANDS     R2, R1, #0X08    ;kolla om Ä2
        BEQ     LOOPN
        B       MAIN

U2      ORR      R2, R1, #0X02    ;hiss upp
        STRb     R2, [R0]
LOOPU   LDRb     R1, [R0]
        ANDS     R2, R1, #0X08    ;kolla om Ä2
        BEQ     LOOPU
        B       MAIN

STOP    B       STOP
;-----
```

END

(14)

```
#include "s3c3410x_asm.h"

; ----- Konstanter -----

START_ADR      EQU      0x00001000
END_ADR        EQU      0x00002FFF
STACK_START    EQU      END_ADR

; ----- Initieringskod -----
CODE32
ORG      0
B        MAIN

ORG      START_ADR
; Initiera stackpekare
MAIN     LDR      SP,=0x2000
         LDR      R4,=TABELL      ;tabelladress till R4
         MOV      R5,#3           ;antal element i tabellen till R5
         BL       ARRAY
STOP     B        STOP

; -----
SWAP     STMFD   SP!, {R3,R5,R6,LR}
         AND     R2,R4,#0xFF      ;maska fram LSB (byte0)
         MOV     R2,R2,LSL#24     ;skifta 24 steg vänster

         MOV     R4,R4,LSR#8
         AND     R3,R4,#0xFF      ;maska fram byte1
         MOV     R3,R3,LSL#16     ;skifta 16 steg vänster

         MOV     R4,R4,LSR#8
         AND     R5,R4,#0xFF      ;maska fram byte2
         MOV     R5,R5,LSL#8      ;skifta 8 steg vänster

         MOV     R4,R4,LSR#8
         AND     R6,R4,#0xFF      ;maska fram byte3

         ORR    R2,R2,R3          ;foga samman
         ORR    R2,R2,R5
         ORR    R2,R2,R6
         LDMFD  SP!, {R3,R5,R6,LR}
         MOV    PC,LR

; -----
ORDER    STMFD   SP!, {R4,LR}
         MOV    R3,R4            ;spara adressvärdet i R3
         LDR    R4,[R4]          ;ladda in 32-bitars tal
         BL    SWAP
         STR    R2,[R3]          ;skriv tillbaka det "swappade"värdet
         LDMFD  SP!, {R4,LR}
         MOV    PC,LR

; -----
ARRAY    STMFD   SP!, {LR}
         CMP    R5,#0            ;kolla om tabellen är tom
         BEQ    EXIT

LOOP     BL     ORDER
         ADD    R4,R4,#4          ;stega fram i tabellen
         SUBS   R5,R5,#1          ;håll koll på antalet element
         BNE   LOOP
EXIT     LDMFD  SP!, {LR}
         MOV    PC,LR

ORG      0x1100
TABELL  DC32    0x89ABCDEF,0x01234567, 0x56789ABC

END
```