ÖVNING 6

Föregås av tre assföreläsningar, exemplen här ska ha lite höjd, tentamensmässiga...

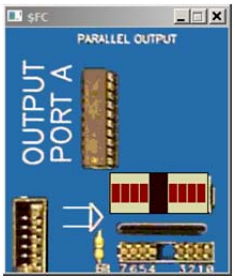
Följande uppgifter demonstreras under övningen:

Exempelsamling:
9.20, 9.26

Självverksamhet:

Exempelsamling:
9.2, 9.6, 9.8, 9.9, 9.11, 9.21, 9.25

9.20 En ramp med ljusdioder, enligt figuren till höger, är ansluten till adress FC_{16} i en FLISP-dator.



a) Skriv en subrutin `BLINK` som får samtliga dioder att blinka genom att först tända och sedan släcka dom. Kontrollera funktionen genom att stega igenom subrutinen instruktionsvis.

b) Utforma, som en ny subrutin `BLINKDELAY`, en fördröjning så att dioderna blinkar även då programmet exekveras normalt.

c) Beskriv subrutinerna `BLINK` och `BLINKDELAY` i form av flödesplaner.

d) Skapa ett enkelt huvudprogram `main`, som kontinueraligt anropar subrutinen `BLINK`.

9.26 I en styrenhet för en maskin används FLIS-datorn. Ett avsnitt av styrprogrammet skall läsa av värdet CASE (8 bitar), som finns på inporten $FB, och därefter utföra en av åtta olika subrutiner, SUB0-SUB7. Vilken subrutin som utförs bestäms av värdet på variabeln CASE. Om CASE = 0 utförs SUB0, om CASE = 1 utförs SUB1 osv. Om CASE > 7 skall ingen av subrutinerna utföras. Adresserna till de åtta olika subrutinerna skall finnas lagrade i minnet i en tabell med början på adressen $C0. Vid laddning av programmet i minnet skall också tabellen med subrutinadresserna laddas. Skriv ett huvudprogram i assemblerspråk som först initierar stackpekaren till $20 och sedan läser värdet som finns på inporten $FB (CASE). Därefter anropas en av subrutinerna enligt ovan om CASE < 8. Programmet utformas som en evighetsslinga. Det skall utformas som flerval och inte som upprepade tvåval.

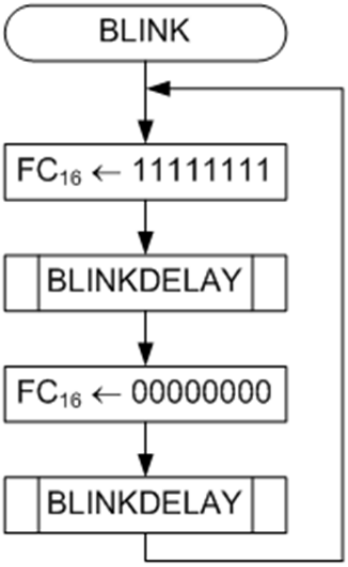
Subrutinernas hexadecimala startadresser skall vara 80, 67, 75, 52, 90, B9, AF och E0.

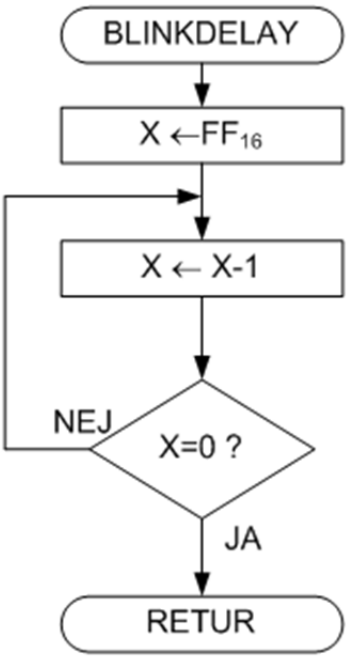
LÖSNINGAR:

9.20

; uppgift a)

```
LedDisplay EQU     $FC
Blink:     LDA     #$FF
           STA
    LedDisplay
           LDA     #0
           STA
    LedDisplay
           BRA     Blink
```





b,c)
modifierad för DELAY:

```
LedDisplay EQU     $FC
Blink:     LDA     #$FF
           STA
    LedDisplay
           JSR
    BLINKDELAY
           LDA     #0
           STA     LedDisplay
           JSR     BLINKDELAY
           BRA     Blink

BLINKDELAY:
           LDX     #$FF
BLINKDELAY1:
           LEAX    -1,X
           CMPX    #0
           BNE     BLINKDELAY1
           RTS
```

d)

```
LedDisplay:EQU     $FC
           ORG     $20
main:      JSR     Blink
           BRA     main
```

9.26

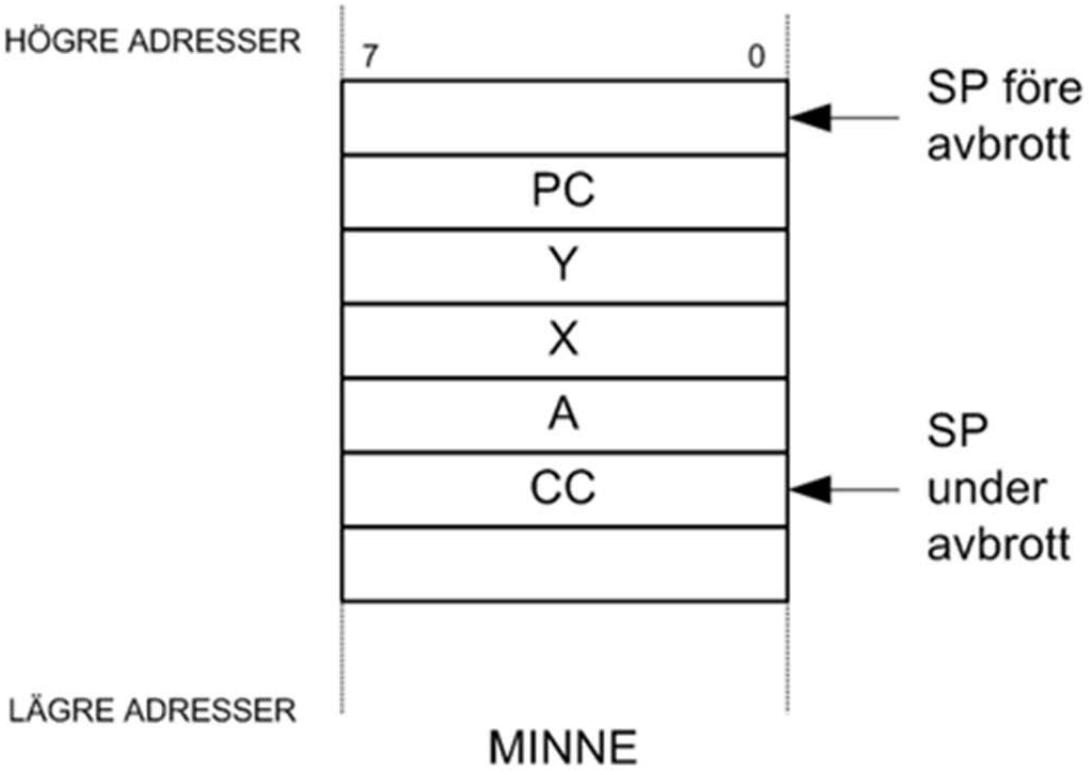
```
BOS:      EQU       $20
DIPSW     :    EQU       $FB
HEXDIS:   EQU       $FB

          ORG       $20
START     :    LDSP      #BOS      ;Initiering av stackpekare
LOOP:          LDX       #JTAB

          LDA       DIPSW          ;Läs inport DIPSW
          CMPA      #7        ;Maxvärde
          BHI       ERROR          ;Ogiltigt. Visa E på HEXDISPLAY

          LDA       A,X       ;Ladda adress till subrutin från tabell
          STA       TEMP
          LDX       TEMP
          JSR       0,X       ;Utför vald subrutin
          BRA       LOOP      ;Upprepa

ERROR     :    LDA       #$E       ;Felkod
          STA       HEXDIS
          BRA       LOOP

TEMP:          RMB       1

JTAB:          FCB       $80,$67,$75,$52,$90,$B9,$AF,$E0
```

00 2	10 3	20 5	30 3	40 3	50 3	60 3	70 3	80 3	90 2	A0 3	B0 3	C0 3	D0 3	E0	F0 2
NOP	PSHA	BSR	STX	STX	STX	STX	STX	STX	LDX	LDX	LDX	LDX	LDX		LDA
ih 1	ih 1	pc 2	ab 2	ns 2	nx 2	ax 1	ny 2	ay 1	im 2	ab 2	ns 2	nx 2	ny 2		im 2
01 4	11 3	21 4	31 3	41 3	51 3	61 3	71 3	81 3	91 2	A1 3	B1 3	C1 3	D1 3	E1 3	F1 3
ANDCC	PSHX	BRA	STY	STY	STY	STY	STY	STY	LDY	LDY	LDY	LDY	LDY	STA	LDA
im 2	ih 1	pc 2	ab 2	ns 2	nx 2	ax 1	ny 2	ay 1	im 2	ab 2	ns 2	nx 2	ny 2	ab 2	ab 2
02 4	12 3	22 4	32 3	42 3	52 3	62 3	72 3	82 3	92 2	A2 3	B2 3	C2 3	D2 3	E2 3	F2 3
ORCC	PSHY	BMI	STSP	STSP	STSP	STSP	STSP	STSP	LDSP	LDSP	LDSP	LDSP	LDSP	STA	LDA
im 2	ih 1	pc 2	ab 2	ns 2	nx 2	ax 1	ny 2	ay 1	im 2	ab 2	ns 2	nx 2	ny 2	ns 2	ns 2
03	13 3	23 4	33 2	43 2	53 4	63 4	73 4	83 4	93 4	A3 5	B3 5	C3 5	D3 5	E3 3	F3 3
	PSHCC	BPL	JMP	RTS	JMP	JMP	JMP	JMP	SBCA	SBCA	SBCA	SBCA	SBCA	STA	LDA
	ih 1	pc 2	ab 2	ih 1	nx 2	ax 1	ny 2	ay 1	im 2	ab 2	ns 2	nx 2	ny 2	nx 2	nx 2
04	14 3	24 4	34 4	44 6	54 5	64 5	74 5	84 5	94 4	A4 5	B4 5	C4 5	D4 5	E4 3	F4 3
	PULA	BEQ	JSR	RTI	JSR	JSR	JSR	JSR	SUBA	SUBA	SUBA	SUBA	SUBA	STA	LDA
	ih 1	pc 2	ab 2	ih 1	nx 2	ax 1	ny 2	ay 1	im 2	ab 2	ns 2	nx 2	ny 2	ax 1	ax 1
05 3	15 3	25 4	35 3	45 3	55 3	65 3	75 3	85 3	95 4	A5 5	B5 5	C5 x	D5 5	E5 4	F5 4
CLRA	PULX	BNE	CLR	CLR	CLR	CLR	CLR	CLR	ADCA	ADCA	ADCA	ADCA	ADCA	STA	LDA
ih 1	ih 1	pc 2	ab 2	ns 2	nx 2	ax 1	ny 2	ay 1	im 2	ab 2	ns 2	nx 2	ny 2	x+ 1	x+ 1
06 3	16 3	26 4	36 4	46 4	56 4	66 4	76 4	86 4	96 4	A6 5	B6 5	C6 5	D6 5	E6 4	F6 4
NEGA	PULY	BVS	NEG	NEG	NEG	NEG	NEG	NEG	ADDA	ADDA	ADDA	ADDA	ADDA	STA	LDA
ih 1	ih 1	pc 2	ab 2	ns 2	nx 2	ax 1	ny 2	ay 1	im 2	ab 2	ns 2	nx 2	ny 2	x- 1	x- 1
07 3	17 3	27 4	37 4	47 4	57 4	67 4	77 4	87 4	97 3	A7 4	B7 4	C7 4	D7 4	E7 4	F7 4
INCA	PULCC	BVC	INC	INC	INC	INC	INC	INC	CMPA	CMPA	CMPA	CMPA	CMPA	STA	LDA
ih 1	ih 1	pc 2	ab 2	ns 2	nx 2	ax 1	ny 2	ay 1	im 2	ab 2	ns 2	nx 2	ny 2	+x 1	+x 1
08 3	18 2	28 4	38 4	48 4	58 4	68 4	78 4	88 4	98 3	A8 4	B8 4	C8 4	D8 4	E8 4	F8 4
DECA	TFR A,C	BCS	DEC	DEC	DEC	DEC	DEC	DEC	BITA	BITA	BITA	BITA	BITA	STA	LDA
ih 1	ih 1	pc 2	ab 2	ns 2	nx 2	ax 1	ny 2	ay 1	im 2	ab 2	ns 2	nx 2	ny 2	-x 1	-x 1
09 2	19 2	29 4	39 3	49 3	59 3	69 3	79 3	89 3	99 4	A9 5	B9 5	C9 5	D9 5	E9 3	F9 3
TSTA	TFR C,A	BCC	TST	TST	TST	TST	TST	TST	ANDA	ANDA	ANDA	ANDA	ANDA	STA	LDA
ih 1	ih 1	pc 2	ab 2	ns 2	nx 2	ax 1	ny 2	ay 1	im 2	ab 2	ns 2	nx 2	ny 2	ny 2	ny 2
0A 3	1A 2	2A 4	3A 4	4A 4	5A 4	6A 4	7A 4	8A 4	9A 4	AA 5	BA 5	CA 5	DA 5	EA 3	FA 3
COMA	TFR X,Y	BHI	COM	COM	COM	COM	COM	COM	ORA	ORA	ORA	ORA	ORA	STA	LDA
ih 1	ih 1	pc 2	ab 2	ns 2	nx 2	ax 1	ny 2	ay 1	im 2	ab 2	ns 2	nx 2	ny 2	ay 1	ay 1
0B 3	1B 2	2B 4	3B 4	4B 4	5B 4	6B 4	7B 4	8B 4	9B 4	AB 5	BB 5	CB 5	DB 5	EB 4	FB 4
LSLA	TFR Y,X	BLS	LSL	LSL	LSL	LSL	LSL	LSL	EORA	EORA	EORA	EORA	EORA	STA	LDA
ih 1	ih 1	pc 2	ab 2	ns 2	nx 2	ax 1	ny 2	ay 1	im 2	ab 2	ns 2	nx 2	ny 2	y+ 1	y+ 1
0C 3	1C 2	2C 4	3C 4	4C 4	5C 4	6C 4	7C 4	8C 4	9C 3	AC 4	BC 4	CC 4	DC 4	EC 4	FC 4
LSRA	TFR X,S	BGT	LSR	LSR	LSR	LSR	LSR	LSR	CMPX	CMPX	CMPX	LEAX	LEAX	STA	LDA
ih 1	ih 1	pc 2	ab 2	ns 2	nx 2	ax 1	ny 2	ay 1	im 2	ab 2	ns 2	nx 2	ns 2	y- 1	y- 1
0D 3	1D 2	2D 4	3D 4	4D 4	5D 4	6D 4	7D 4	8D 4	9D 3	AD 4	BD 4	CD 4	DD 4	ED 4	FD 4
ROLA	TFR S,X	BGE	ROL	ROL	ROL	ROL	ROL	ROL	CMPY	CMPY	CMPY	LEAY	LEAY	STA	LDA
ih 1	ih 1	pc 2	ab 2	ns 2	nx 2	ax 1	ny 2	ay 1	im 2	ab 2	ns 2	ny 2	ns 2	+y 1	+y 1
0E 3	1E 2	2E 4	3E 4	4E 4	5E 4	6E 4	7E 4	8E 4	9E 3	AE 4	BE 4	CE 4	DE 4	EE 4	FE 4
RORA	TFR Y,S	BLE	ROR	ROR	ROR	ROR	ROR	ROR	CMPSP	CMPSP	LEASP	LEASP	LEASP	STA	LDA
ih 1	ih 1	pc 2	ab 2	ns 2	nx 2	ax 1	ny 2	ay 1	im 2	ab 2	ns 2	nx 2	ny 2	-y 1	-y 1
0F 3	1F 2	2F 4	3F 4	4F 4	5F 4	6F 4	7F 4	8F 4	9F 4	AF 4	BF 4	CF 4	DF	EF	FF
ASRA	TFR S,Y	BLT	ASR	ASR	ASR	ASR	ASR	ASR	EXG A,C	EXG X,Y	EXG X,S	EXG Y,S			
ih 1	ih 1	pc 2	ab 2	ns 2	nx 2	ax 1	ny 2	ay 1	ih 1	ih 1	ih 1	ih 1			

Vektorer och I/O

$FF ”Reset”, återstart
$FE ”Exception”, undantag för ej implementerad operationskod
$FD ”IRQ”, avbrott
$FC ”I/O” Port för anslutning av perifer enhet
$FB ”I/O” Port för anslutning av perifer enhet

Stackens utseende vid avbrott/undantag



Vid "IRQ" pekar PC på den instruktion som skulle ha utförts om inget avbrott signalerats.

Vid "Exception" pekar PC på instruktionen omedelbart efter den otillåtna operationskoden.