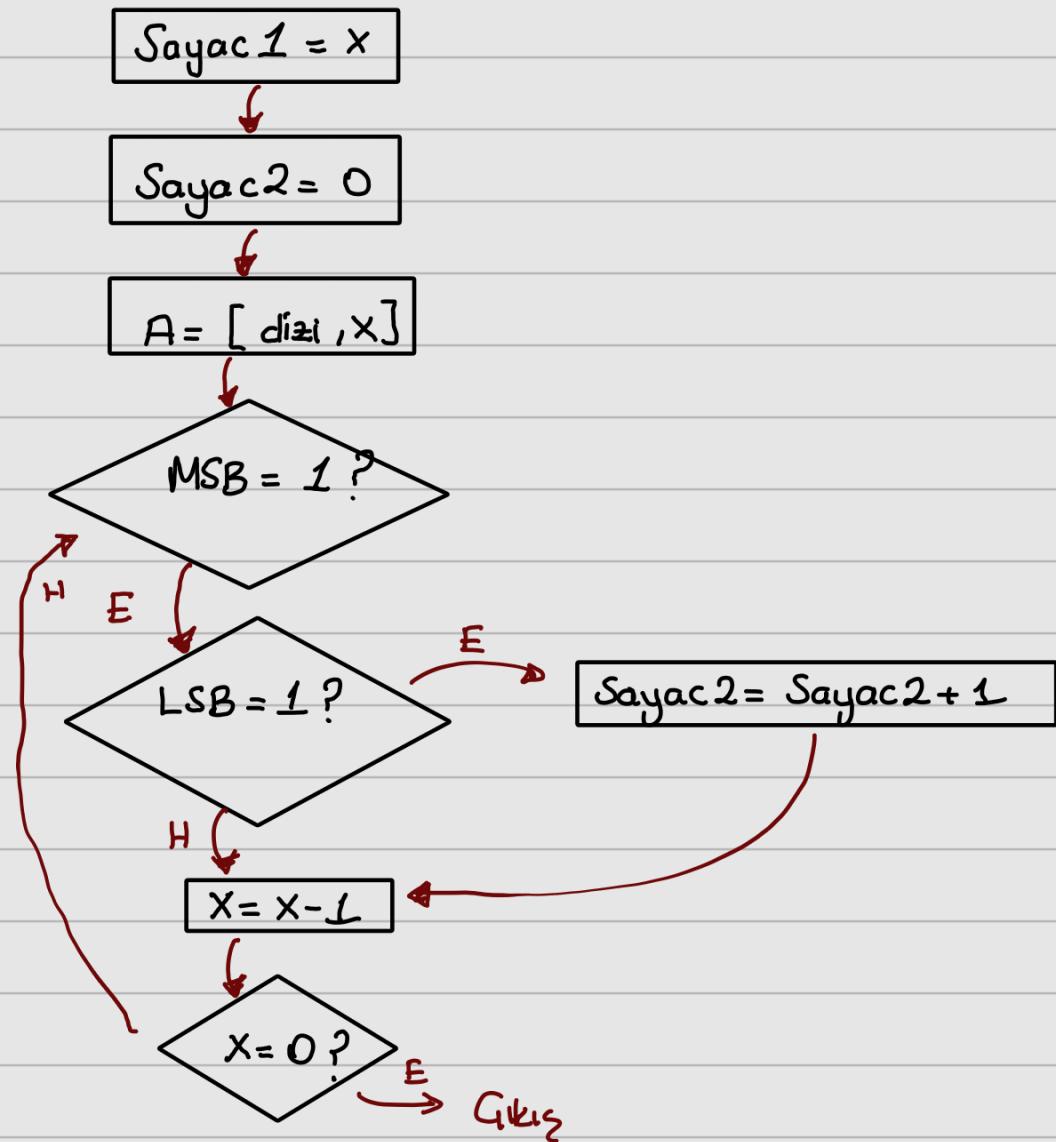


Hafta 11: Zaman Geciktirme İşlemleri ve Alt yordan Kavramı

Kitaptaki Örnek 8: LSB ve MSB bitleri 1 olanların toplamını bul?



• code 0200

0200 : A4 00	basla	LDY #\$'00 ; biriktirme yerini sıfırla
0202 : A2 05		LDX #\$'05 : dizi = \$'05
0204 : BS 1F	yeni	LDA \$'1F,X ; diziden veri al.
0206 : 30 04		BMI sor ; sor, bir mi ?
0208 : C A		DEX ; hayır, X'i bir azalt.
0209 : D0 F9		BNE yeni ; bitmediyse devam.
020B : 00		BRK

020 C : 6A sor ROR ; evet, lsb'ye bak
 020 D : 90 01 BCC öteki ; 0 ise digerine bak.
 020 F : C8 INY ; elde var bir.

0210 : CA öteki DEX ; orjinal sayaci bir azalt.
 0211 : D0F1 BNE yeni ; azalt devam et.
 0213 : 00 bitti BRK

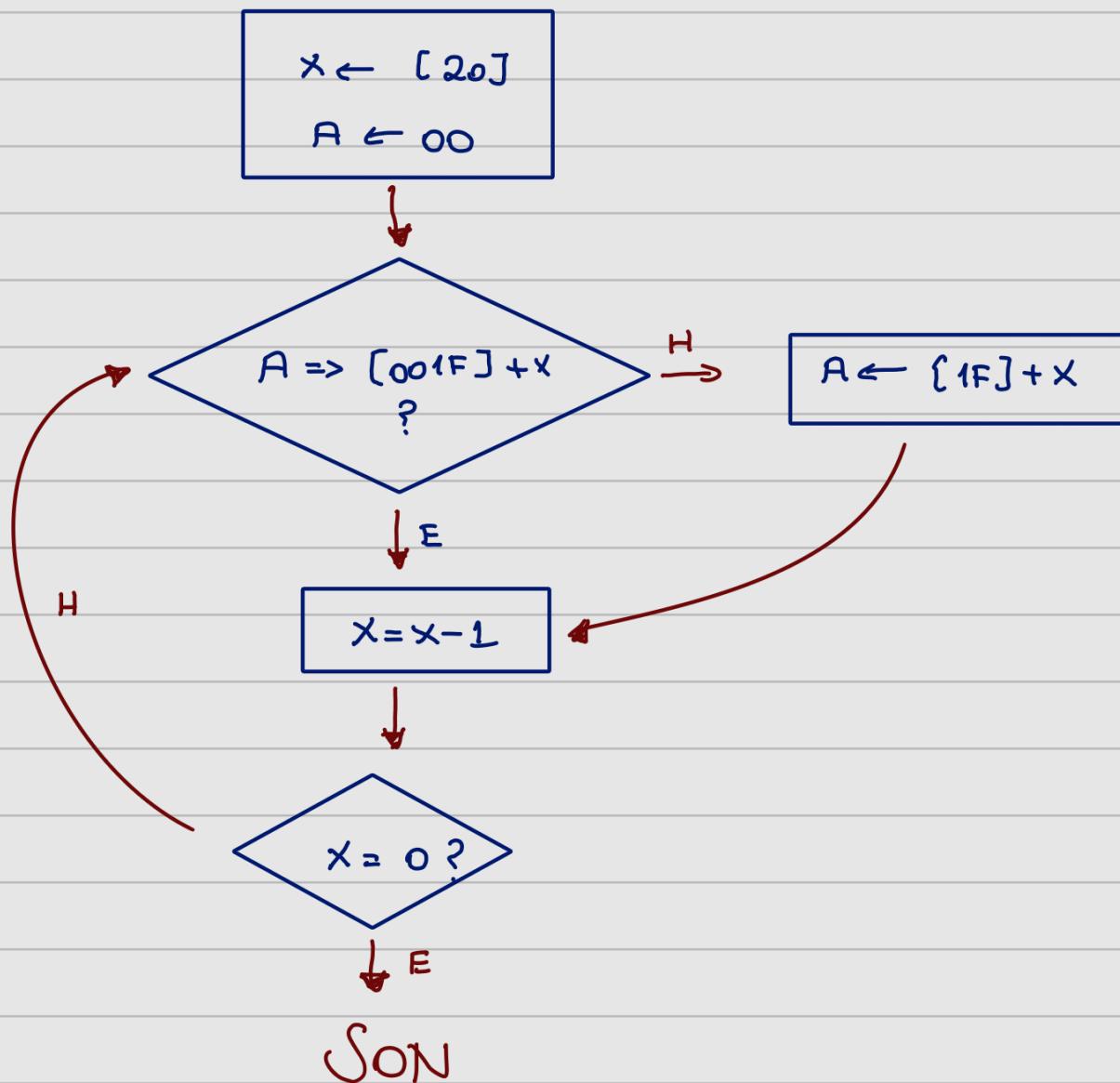
sinavda böyle gelir mesela, pozitif olan, 4-bit
 1 olan vs vs. gelebilir.

basla LDY #\$00 ; sayac = 0
 LDX #\$05 ; dizide 5 eleman var.

yeni LDA \$1F,X ; diziden veri al
 BMI sor ; MSB 1'e sor a git
 JMP öteki ; MSB 0 ise direkt öteki
 DEX-BNS 5 cycle je git.

↴ sor ROR ; LSB carrye al.
 daha fazla BCC öteki ; LSB 0'sa sayma
 kommt INY ; 1'ye say.
 dmasma
 rägmen
 daha hızlı . öteki DEX ; sonraki elemen geq.
 BNE yeni ; $X \neq 0$ döngüye devam.
 bitti BRK

Örnek 9: Bellekteki sırasız bir grup veri içerisinde en büyük değerlikli elemanların bulunması.



SON

- data \$20
- byte \$05, \$75, \$8D, \$E0, \$99, \$24
- code \$0200

basla

LDX \$20 ; $X \leftarrow$ eleman sayisi (5)

LDA #\\$00 ; $A \leftarrow 0$ baslangicta en küçük de.

yeni

CMP \$20,X ; A ile Dizi $[x]$ karşlaştır.

Eğer $A > M$ ise

$C=1$, Değilse $C=0$

BCS Max ; $A \geq Dizi[X]$ ise \rightarrow büyük olan A' 'ya
LDA \$20,X ; değilse yeni büyük A' 'yı al.

Max DEX ; $X \leftarrow X-1$ (bir önceki elemenin
BNE yeni ; $X \neq 0 \rightarrow$ yeni elemenle devam et.

STA \$80 ; en büyük değeri \$80 ad. sakla.
bitti BRK

7.5 Zaman Geciktirme

Mikrois. bazı işlemleri belirli bir süre beklenerek
yapılmalıdır.

- Sensörden veri okuma aralıkları
- LED yak söndür işlemleri
- Seri haberleşme gecikmeleri
- Motor kontrol zamanlaması

Bu tür programlar için geciktirici alt prog.
yazılır.

Cycle hesabı önemlidir. 6502 de her cycle
bilinir.

Gecikme süresi = cycle \times döngü sayısı

i) İndis kaydediciler bas. deg. sayac yükle.

ii) Geçikme komut satırlarını ıgle

iii) Sayacı azalt.

iv) 2 ve 3. adım tekrarı.

BASLA LDX #\$01 ; sayac bilgisi yükle \rightarrow 2 cycle

DON NOP ; Don \rightarrow 2 cycle

DEX ; X azalt \rightarrow 2 cycle

BNE DON ; 0 degilse dön. BNe

RTS
branch alınsa \rightarrow 3
alınmazsa \rightarrow 2

Toplam Geçikme = $T_d + (\text{Sayac} \times T_i) - 1$
 \downarrow Döngü dışı \downarrow döngü içi

RTS \rightarrow 6 cycle

$$2 + (01 * ((2+2+3)-1)) = 8 \mu s \text{ (8 mikrosaniye)}$$

8 , RTS döngü içi olmadığı için

$$\underline{8 + 6 = 14 \text{ mikrosaniye}}$$

İki Döngülü Gecikme (x ve y kullanımı)

Daha uzun sürelerde ihtiyaç duyulunca kullanılır.

LDX sayac1

LDY sayac2

BEK1 DEX

BNE BEK1 ; x bitti mi?

BEK2 DEY

BNE BEK2 ; y bitti mi?

RTS

$$T_g = T_d + ((\text{Sayac1} * \tau_i) - 1) + ((\text{Sayac2} * \tau_i) - 1)$$

$$T_g = (3+3) + ((\text{Sayac1} * (2+3))-1) + ((\text{Sayac2} * 2+3)-1)$$

$$T_{g\max} = 6 + ((255 * 5) - 1) + ((255 * 5) - 1)$$

İç içe (Gopraz) Döngü Gecikmesi

GEC1 LDX \$20

LDY \$21

BEK DEX

BNE BEX

DEY

BNE BEK

RTS

$$T_g = \text{LDX}(3) + \text{LDY}(3) + \text{DEX}(2) + \text{BNE}(2) + \text{DEY}(2) + \text{BNE}(2) + \text{RTS}(6)$$