***Nhat Doan-Chris Grove - CSC137- Homework 2***

**5-13>**

SUB:

*D1T4: DR ← M[AR]*

*D1T5: DR ← AR, AR← DR*

*D1T6: AC ← (AC*

*D1T7: AC ← AC + 1*

*D1T8: AC ← AC + DR, SC ← 0*

XCH:

*D1T4: DR ← M[AR]*

*D1T5: M[AR] ← AC, AC← DR, SC ← 0*

**5-21>**

Control Gate associated with PC, We assume those signals

*A = { 1 : when DR = 0 and*

*0: otherwise }*

*M=AC (15)*

*N=1 if AC =0*

*INC = R’T1+ RT2+ AD6T6+ I’D7T3 [M’ B4+ MB3+N’B2+E’B1]+ ID7T3[B9(FGI)+B8(FGO)]*

*CLR=RT1*

*LD = D4T4 +D5T5 + …*

**5-22>**

Control gates for the write input of the memory, we need to look for the destination where is says M[AR]

*WRITE = RT1 + D3T4+D5T4+ D6T6*

**5.24>**

Control gates for boolean of x2, we need to look at table 5-6 where the source is PC because x2 = 1 is PC.

R’T0: AC ← PC

RT0: AR ← 0 , TR ← PC

D5T4: M[AR] ← PC, AR ← AR + 1

⇒ *x2= R’T0+ RT0 + D5T4 = T0( R+R’) + D5T4 = T0 + D5T4.*

**1>** Fetch: R’T0: LD(AR), x2

R’T1: LD(IR), INC(PC) , READ, x7

Decode: at R’T2: LD(AR), LD(I), x5

Indirect: at D7’IT3: LD(AR), READ, x7

List of signals:

-LD(AR): R’T0 + R’T2 + D7’IT3

-LD(I): R’T2

-LD(IR): R’T1

-INC(PC): R’T1

-READ: R’T1 + D7’IT3

-x2: R’T0

-x5: R’T2

-x7: R’T1+ D7’IT3

**2>** Memory- reference :

AND: at D0T4: LD(DR) , READ, x7

At D0T5: LD(AC),CLR(SC)

ADD: at D1T4: LD(DR), READ, x7

at D1T5: LD(AC),LD(E), CLR(SC)

LDA: at D2T4: LD(DR), READ , x7

at D2T5 : LD(AC),x3,CLR(SC)

STA: at D3T4: WRITE, x4, CLR(SC)

BUN: at D4T4: LD(PC), x1, CLR(SC)

BSA: at D5T4: INC(AR),WRITE, x2

at D5T5: LD(PC), x1, CLR(SC)

ISZ: at D6T4: LD(DR), READ, x7

at D6T5: INC(DR)

At D6T6: WRITE,if(DR=0) INC(PC), x3, CLR(SC)

CLR(SC) :D0T5 +D1T5 + D2T5+D3T4+ D4T4+D5T5+D6T6

LD(AC): D0T5 + D1T5 + D2T5

LD(DR): D0T4+ D1T4+ D2T4+ D6T4

LD(E): D1T5

LD(PC)(if DR=0) : D4T4 +D5T5

INC(AR): D5T4

INC(DR): D6T5

INC(PC): D6T6

READ: D0T4+D1T4+D2T4+D6T4

WRITE: D3T4+D5T4+ D6T6

X1: D4T4+ D5T5

X2: D5T4

X3: D2T5+D6T6

X4: D3T4

X7: D0T4 + D1T4 + D2T4 +D6T4

**3>** interrrupt\_handling is an open communication only when some data has to be passed ( Input/ Output) When the interface found that the I/O device is ready for transfer some data, it generates an interrupt request to the CPU

Interrupt-Handling is useful because it can help transfer data in and out without being controlled by the clocked. Let say when a key is pressed, there is an interrupt request sent to the CPU to get the key information. Therefore, interrupt handlers can be used to control the flow of signals in the computer system.