Fetch R’T0: AR🡨PC

R’T1: IR🡨M[AR], PC🡨PC+1

Decode R’T2: D0......D7🡨Decode IR(12-14),AR🡨IR(0-11), I🡨IR(15)

INDIRECT D’7IT3: AR🡨M[AR]

Interrupt:

T0’T1’T2’(IEN)(FGI+FGO): R🡨1

RT0: AR🡨0,TR🡨PC

RT1: M[AR]🡨TR, PC🡨0

RT2: PC🡨PC+1, IEN🡨0, R🡨0,SC🡨0

Memory-Reference:

AND D0T4: DR🡨M[AR]

D0T5: AC🡨AC^DR, SC🡨0

ADD D1T4: DR🡨M[AR]

D1T5: AC🡨AC+DR, C🡨COUT, SC🡨0

LDA D2T4: DR🡨M[AR]

D0T4: AC🡨DR, SC🡨0

STA D3T4: M[AR]🡨AC, SC🡨0

BUN D4T4: PC🡨AR, SC🡨0

BSA D5T4: M[AR]🡨PC, AR🡨AR+1

D5T5­: PC🡨AR, SC🡨0

ISZ D6T4: DR🡨M[AR]

D6T5: DR🡨DR+1

D6T6: M[AR]🡨DR, If(DR==0) then (PC🡨PC+1)

Resister-Reference:

IR(i)=Bi(i=0,1,2,3,….,11)

CLA D7I’T3B11: AC🡨0, SC🡨0

CLE D7I’T3B11: E🡨0, SC🡨0

CMA D7I’T3B11: AC🡨AC’, SC🡨0

CME D7I’T3B11: E🡨E’, SC🡨0

CIR D7I’T3B7: AC🡨shr AC, AC(15)🡨E, E🡨AC(0) , SC🡨0

CIR D7I’T3B6: AC🡨shl AC, AC(0)🡨E, E🡨AC(15) , SC🡨0

INC D7I’T3B5: AC🡨AC+1, SC🡨0

SPA D7I’T3B4: If(AC(15)==0)then(PC🡨PC+1) , SC🡨0

SNA D7I’T3B3: If(AC==1)then(PC🡨PC+1) , SC🡨0

SZA D7I’T3B2: If(AC==0)then(PC🡨PC+1) , SC🡨0

SZE D7I’T3B1: If(E==0)then(PC🡨PC+1) , SC🡨0

HLT D7I’T3B0: S🡨0, SC🡨0

Input-Output:

INP D7IT3B11: AC(0-7)🡨INPR, FGI🡨0, SC🡨0

OUT D7IT3B10: OUTR🡨AC(0-7), FGO🡨0, SC🡨0

SKI D7IT3B9: If(FGI==1)then (PC🡨PC+1) , SC🡨0

SKO D7IT3B8: If(FGO==1)then (PC🡨PC+1) , SC🡨0

ION D7IT3B7: IEN🡨1, SC🡨0

IOF D7IT3B6: IEN🡨0, SC🡨0