

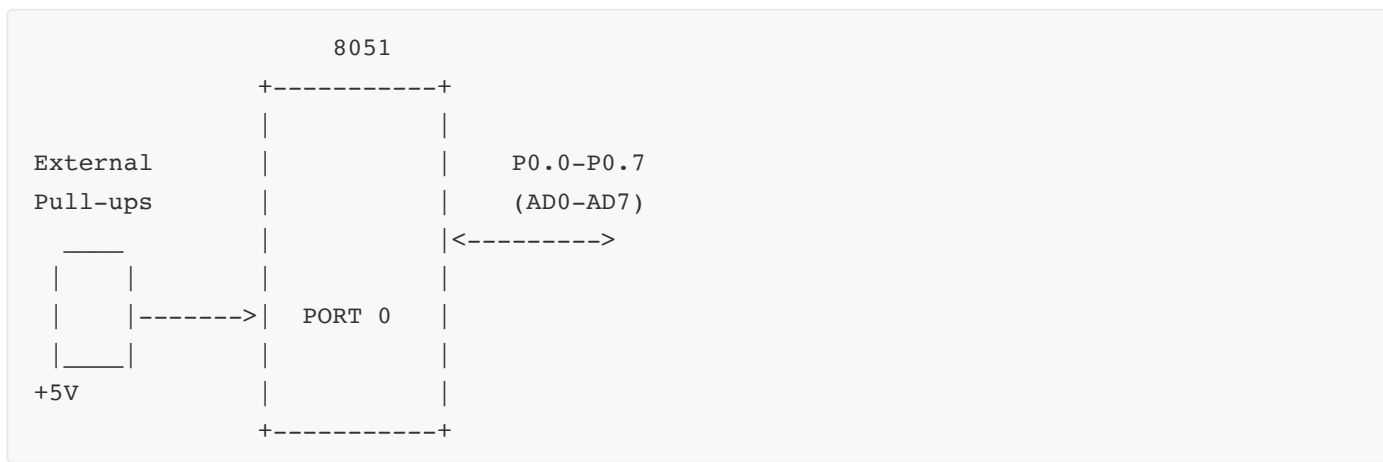
## પ્રશ્ન 1(અ) [3 ગુણ]

8051 માઇક્રોકંટ્રોલરના કોઈપણ એક પોર્ટ કન્ફિગરેશનનું વર્ણન કરો.

જવાબ:

કન્ફિગરેશન	વર્ણન
પોર્ટ 0	ડ્યુઅલ-પર્પઝ પોર્ટ - 8-બિટ ઓપન ડ્રેન બિડાયરેક્શનલ I/O પોર્ટ અને મલ્ટીપ્લેક્સ લો એડ્રેસ/ડેટા બસ. I/O ફંક્શન માટે બાહ્ય પુલ-અપ રેસિસ્ટર જરૂરી.

સાચાગ્રામ:



મેમરી ટ્રીક: "પોર્ટ 0-પ્લેસ" (પોર્ટ 0 ને પુલ-અપ્સ જોઈએ, લેચ/એડ્રેસ/ડેટા તરીકે કામ કરે)

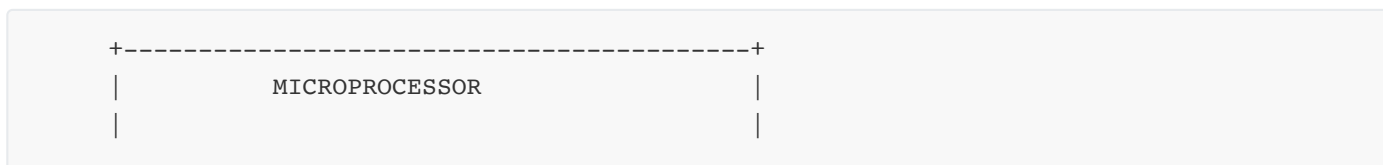
## પ્રશ્ન 1(બ) [4 ગુણ]

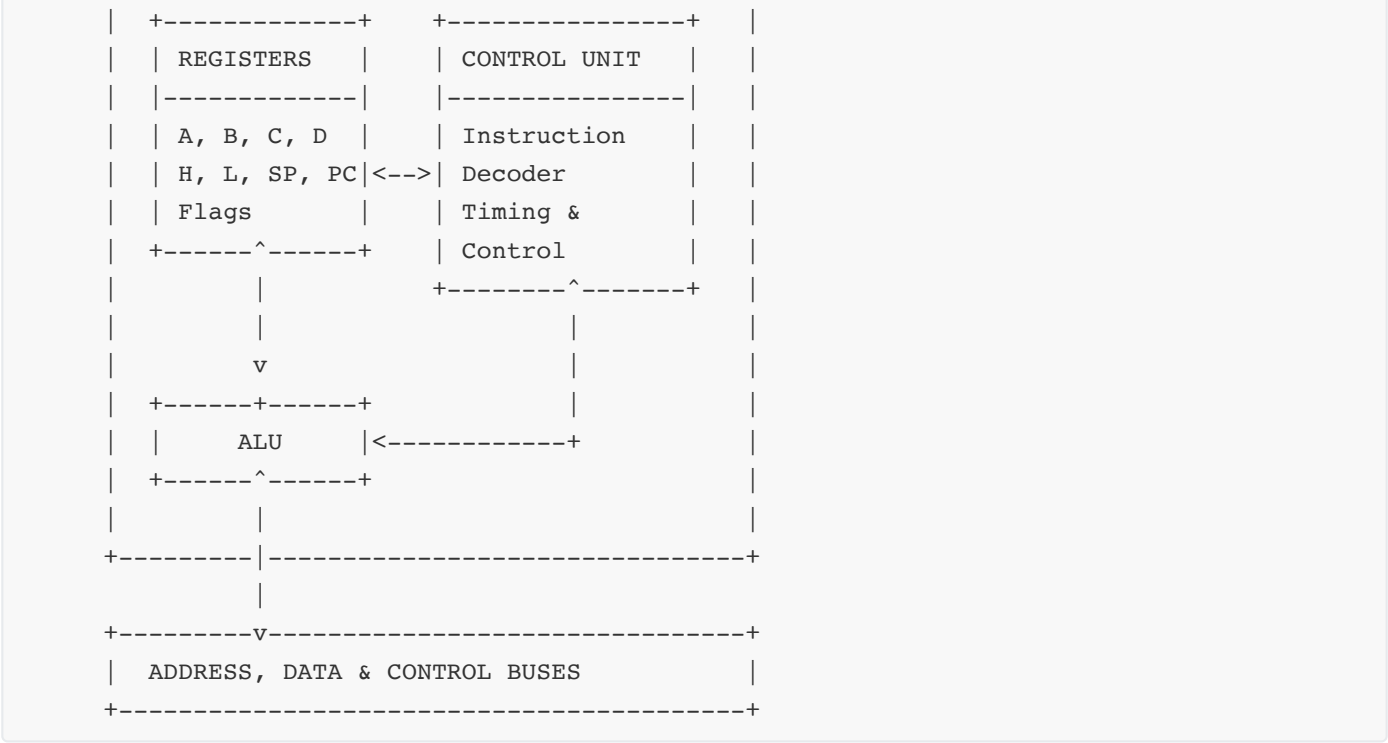
માઇક્રોપ્રોસેસર આર્કિટેક્ચરનું વર્ણન કરો.

જવાબ:

ઘટક	કાર્ય
ALU	ગાણિતિક અને લોજિકલ ઓપરેશન કરે છે
રજિસ્ટર્સ	ડેટા અને એડ્રેસ માટે કામચલાઉ સ્ટોરેજ
કંટ્રોલ યુનિટ	પ્રોસેસર ઓપરેશન અને ડેટા ફ્લો નિર્દેશિત કરે છે
બસ	ડેટા ટ્રાન્સફર માટે પાથવે (એડ્રેસ, ડેટા, કંટ્રોલ)

સાચાગ્રામ:





મેમરી ટ્રીક: "RABC" - "રજિસ્ટર, ALU, બસ, કંટ્રોલ"

પ્રશ્ન 1(ક) [7 ગુણ]

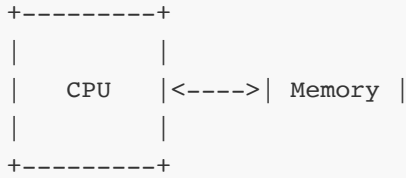
વોન ન્યુમેન અને હાર્વર્ડ આર્કિટેક્ચરની તુલના કરો.

જવાબ:

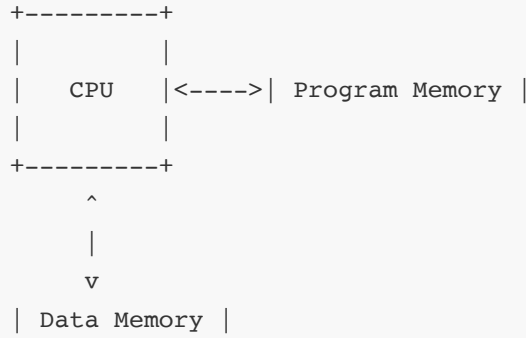
ફીચર	વોન ન્યુમેન આર્કિટેક્ચર	હાર્વર્ડ આર્કિટેક્ચર
મેમરી બસ	ઇન્સ્ટ્રક્શન અને ડેટા માટે એક જ મેમરી બસ	પ્રોગ્રામ અને ડેટા મેમરી માટે અલગ બસ
એક્ઝિક્યુશન	સિક્વેન્શિયલ એક્ઝિક્યુશન	પેરેલલ ફેચ અને એક્ઝિક્યુટ શક્ય
સ્પીડ	બસ બોટલનેક ને કારણે ધીમું	સમાંતર એક્સેસને કારણે ઝડપી
મેમરી એક્સેસ	એક જ મેમરી સ્પેસ	અલગ મેમરી સ્પેસ
જટિલતા	સરળ ડિઝાઇન	વધુ જટિલ ડિઝાઇન
ઉપયોગો	સામાન્ય કમ્પ્યુટિંગ	DSP, માઇક્રોકન્ટ્રોલર, એમ્બેડેડ સિસ્ટમ
ઉદાહરણો	મોટાભાગના PC, 8085, 8086	8051, PIC, ARM Cortex-M

સાચાગ્રામ:

Von Neumann:



Harvard:



**મેમરી ટ્રીક:** "હાર્વર્ડ હંમેશા અલગ રસ્તા રાખે" (હાર્વર્ડમાં મેમરી પાથ અલગ હોય છે)

## પ્રશ્ન 1(ક OR) [7 ગુણ]

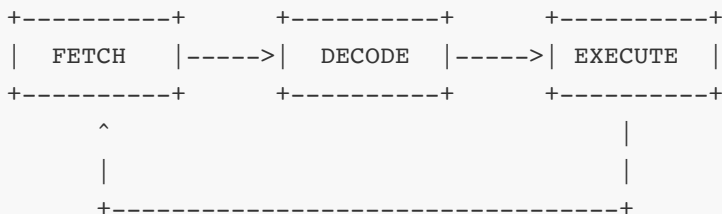
RISC, CISC, Opcode, Operand, Instruction Cycle, Machine Cycle, અને T State ને વ્યાખ્યાયિત કરો.

**જવાબ:**

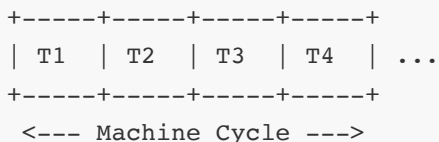
શબ્દ	વ્યાખ્યા
RISC	રિડ્યુસ્ડ ઇન્સ્ટ્રક્શન સેટ કમ્પ્યુટર - સરળ ઇન્સ્ટ્રક્શન સાથે સ્પીડ માટે ઓપ્ટિમાઇઝ્ડ આર્કિટેક્ચર
CISC	કોમ્પ્લેક્સ ઇન્સ્ટ્રક્શન સેટ કમ્પ્યુટર - જટિલ, શક્તિશાળી ઇન્સ્ટ્રક્શન સાથેનું આર્કિટેક્ચર
Opcode	ઓપરેશન કોડ - ઇન્સ્ટ્રક્શનનો ભાગ જે કયા ઓપરેશન કરવાના છે તે સ્પષ્ટ કરે છે
Operand	ઓપરેશનમાં વપરાતો ડેટા વેલ્યુ અથવા એડ્રેસ
Instruction Cycle	ઇન્સ્ટ્રક્શન ફેચ, ડિકોડ અને એક્ઝિક્યુટની સંપૂર્ણ પ્રક્રિયા
Machine Cycle	મૂળભૂત ઓપરેશન જેમ કે મેમરી રીડ/રાઈટ (ઇન્સ્ટ્રક્શન સાયકલનો ભાગ)
T-State	ટાઈમ સ્ટેટ - પ્રોસેસરમાં સમયનો સૌથી નાનો એકમ (ક્લોક પીરિયડ)

**સાચાગ્રામ:**

Instruction Cycle:



T-States within Machine Cycle:



મેમરી ટ્રીક: "RICO ITEM" (RISC, CISC, Opcode, Instruction cycle, T-state, Execute, Machine cycle)

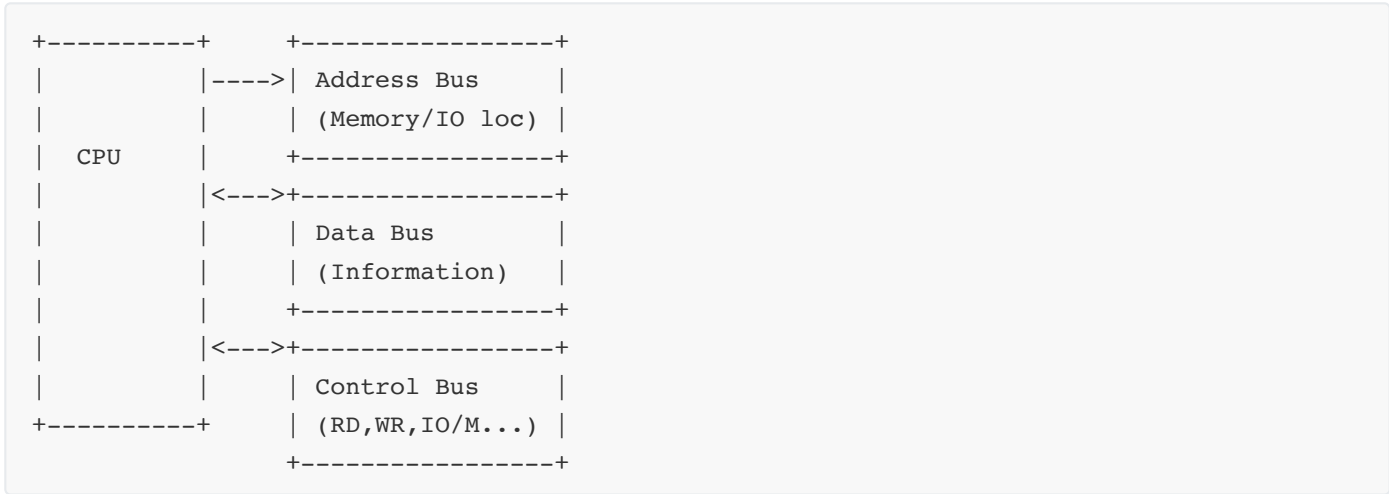
## પ્રશ્ન 2(અ) [3 ગુણ]

ડેટા બસ, એડ્રેસ બસ અને કંટ્રોલ બસ વ્યાખ્યાયિત કરો.

જવાબ:

બસ પ્રકાર	વ્યાખ્યા
ડેટા બસ	બિડાયરેક્શનલ પાથવે જે માઇક્રોપ્રોસેસર અને પેરિફેરલ ડિવાઇસ વચ્ચે વાસ્તવિક ડેટા ટ્રાન્સફર કરે છે
એડ્રેસ બસ	યુનિડાયરેક્શનલ પાથવે જે એક્સેસ કરવાના મેમરી/IO ડિવાઇસ લોકેશન ધરાવે છે
કંટ્રોલ બસ	સિગ્નલ લાઈનોનો ગ્રુપ જે સિસ્ટમ ઓપરેશનને કોઓર્ડિનેટ અને સિન્ક્રોનાઇઝ કરે છે

ડાયાગ્રામ:



મેમરી ટ્રીક: "ADC" - "એડ્રેસ લોકેશન શોધે, ડેટા માહિતી લઈ જાય, કંટ્રોલ ઓપરેશન કોઓર્ડિનેટ કરે"

## પ્રશ્ન 2(બ) [4 ગુણ]

માઇક્રોપ્રોસેસર અને માઇક્રોકન્ટ્રોલરની સરખામણી કરો.

જવાબ:

ફીચર	માઇક્રોપ્રોસેસર	માઇક્રોકન્ટ્રોલર
વ્યાખ્યા	એકલ ચિપ પર CPU	એકલ ચિપ પર સંપૂર્ણ કમ્પ્યુટર સિસ્ટમ
મેમરી	બાહ્ય RAM/ROM જરૂરી	અંદર જ RAM/ROM
I/O પોર્ટ	મર્યાદિત અથવા ચિપ પર નથી	ચિપ પર ઘણા I/O પોર્ટ
પેરિફેરલ્સ	બાહ્ય પેરિફેરલ્સ જરૂરી	અંદર જ પેરિફેરલ્સ (ટાઇમર્સ, ADC, વગેરે)
ઉપયોગો	સામાન્ય કમ્પ્યુટિંગ, PC	એમ્બેડેડ સિસ્ટમ, IoT ડિવાઇસિસ
કિંમત	સંપૂર્ણ સિસ્ટમ માટે વધારે	ઓછી (ઓલ-ઇન-વન સોલ્યુશન)
પાવર કન્ઝમ્પશન	વધારે	ઓછું

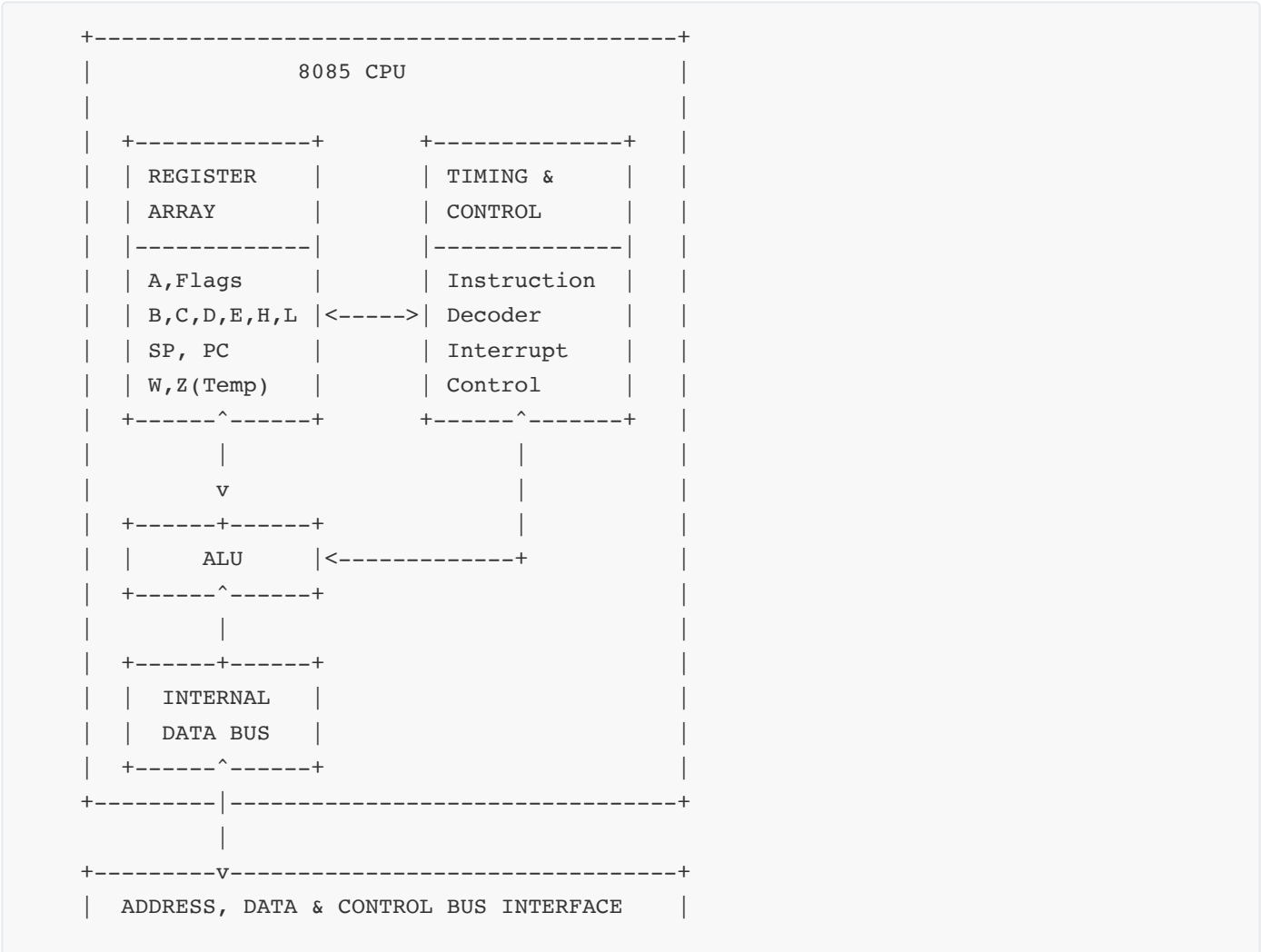
**મેમરી ટ્રીક:** "MEMI-CAP" (મેમરી બાહ્ય/આંતરિક, કિંમત, એપ્લિકેશન્સ, પેરિફેરલ્સ)

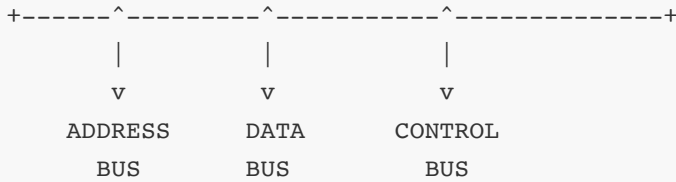
પ્રશ્ન 2(ક) [7 ગુણ]

8085 બ્લોક ડાયાગ્રામ સ્કેચ કરો અને સમજાવો.

જવાબ:

ડાયાગ્રામ:





### મુખ્ય ઘટકો:

- **રજિસ્ટર અેરે:** A (એક્યુમ્યુલેટર), ફ્લેગ્સ, B-L, SP, PC, ટેમ્પ રજિસ્ટર્સ
- **ALU:** ગાણિતિક અને લોજિકલ ઓપરેશન કરે છે
- **ટાઈમિંગ & કંટ્રોલ:** કંટ્રોલ સિગ્નલ્સ જનરેટ કરે છે, ઇન્ટરપ્ટ હેન્ડલ કરે છે
- **બસ ઇન્ટરફેસ:** CPU ને બાહ્ય ડિવાઇસ સાથે જોડે છે
- **ઇન્ટરનલ ડેટા બસ:** આંતરિક ઘટકોને જોડે છે

મેમરી ટ્રીક: "RATBI" - "રજિસ્ટર્સ, ALU, ટાઈમિંગ, બસ, ઇન્ટરફેસ"

## પ્રશ્ન 2(અ OR) [3 ગુણ]

એક્યુમ્યુલેટર, પ્રોગ્રામ કાઉન્ટર અને સ્ટેક પોઇન્ટર સમજાવો.

જવાબ:

રજિસ્ટર	કાર્ય
એક્યુમ્યુલેટર (A)	8-બિટ રજિસ્ટર જે ગાણિતિક અને લોજિકલ ઓપરેશનના પરિણામો સ્ટોર કરે છે
પ્રોગ્રામ કાઉન્ટર (PC)	16-બિટ રજિસ્ટર જે આગલા એક્ઝિક્યુટ થનાર ઇન્સ્ટ્રક્શનનું એડ્રેસ રાખે છે
સ્ટેક પોઇન્ટર (SP)	16-બિટ રજિસ્ટર જે મેમરીમાં સ્ટેકના વર્તમાન ટોપને પોઇન્ટ કરે છે

સાચાગ્રામ:

Accumulator:	Program Counter:	Stack Pointer:
+-----+	+-----+	+-----+
A   <--->	PC   <--->	SP   <--->
+-----+	+-----+	+-----+
Data operations	Points to next instruction	Points to stack top
		v
		+-----+
		Stack
		Memory
		+-----+

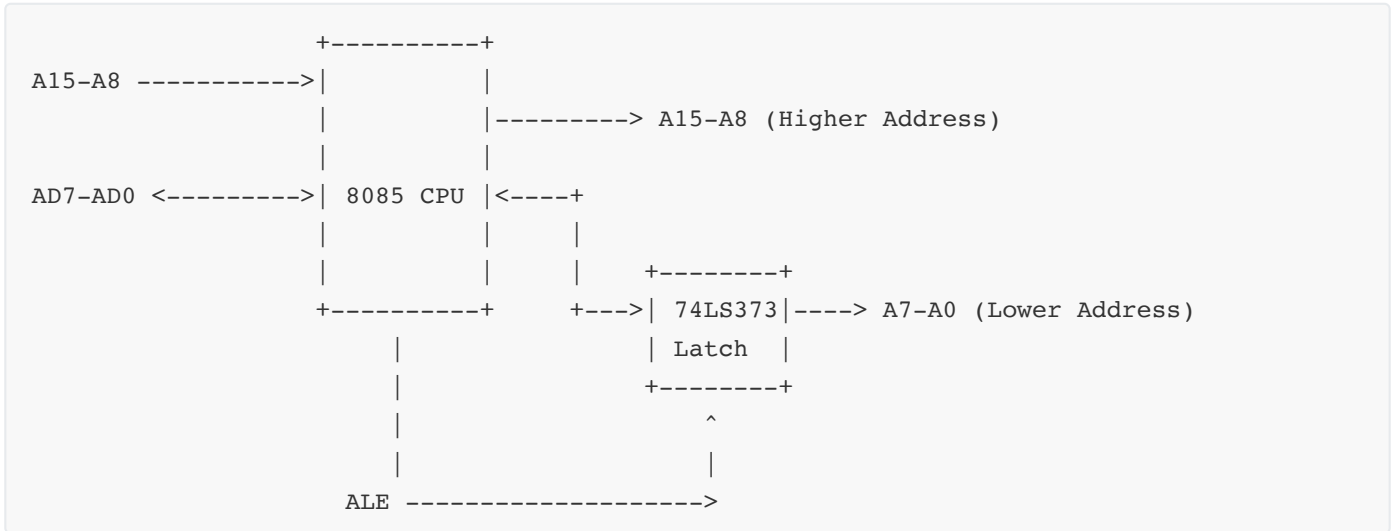
મેમરી ટ્રીક: "APS" - "એક્યુમ્યુલેટર પ્રોસેસ કરે, PC આગલું ઇન્સ્ટ્રક્શન જુએ, SP સ્ટેક સંભાળે"

## પ્રશ્ન 2(બ OR) [4 ગુણ]

એડ્રેસ બસ અને ડેટા બસનું ડિમલ્ટિપ્લેક્સીંગ સ્કેચ કરો અને સમજાવો.

જવાબ:

## સાચાગ્રામ:



## પ્રક્રિયા:

- મલ્ટિપ્લેક્સિંગ:** પિન કાઉન્ટ ઘટાડવા માટે AD0-AD7 પિન એડ્રેસ અને ડેટા સિગ્નલ શેર કરે છે
- ડિમલ્ટિપ્લેક્સિંગના સ્ટેપ્સ:**
  - CPU AD0-AD7 પિન પર એડ્રેસ મૂકે છે
  - ALE (એડ્રેસ લેચ એનેબલ) સિગ્નલ HIGH થાય છે
  - બાહ્ય લેચ (74LS373) લોઅર એડ્રેસ બિટ્સ પકડે છે
  - ALE LOW થાય છે, એડ્રેસ લેચ થઈ જાય છે
  - AD0-AD7 પિન હવે ડેટા લઈ જાય છે

**મેમરી ટ્રીક:** "ALAD" - "ALE એક્ટિવ, લેચ એડ્રેસ, આફ્ટર ડેટા"

## પ્રશ્ન 2(ક OR) [7 ગુણ]

8085 ની કોઈપણ સાત વિશેષતાઓની યાદી આપો.

## જવાબ:

વિશેષતા	વર્ણન
8-બિટ ડેટા બસ	8 બિટ્સ ડેટા પેરેલલમાં ટ્રાન્સફર કરે છે
16-બિટ એડ્રેસ બસ	64KB સુધીની મેમરી એડ્રેસ કરી શકે છે ( $2^{16}$ )
હાર્ડવેર ઇન્ટરપ્ટ	5 હાર્ડવેર ઇન્ટરપ્ટ (TRAP, RST 7.5, 6.5, 5.5, INTR)
સિરિયલ I/O	સિરિયલ કમ્યુનિકેશન માટે SID અને SOD પિન
ક્લોક જનરેશન	ક્રિસ્ટલ સાથે ઓન-ચિપ ક્લોક જનરેટર
ઇન્સ્ટ્રક્શન સેટ	74 ઓપરેશન કોડ્સ જે 246 ઇન્સ્ટ્રક્શન જનરેટ કરે છે
રજિસ્ટર સેટ	છ 8-બિટ રજિસ્ટર (B,C,D,E,H,L), એક્યુમ્યુલેટર, ફ્લેગ્સ, SP, PC

સાચાગ્રામ:

8085 Features			
+-----+			
	+-----+	+-----+	
	8-bit	5 HW	
	Data	Interrupt	
	Bus	Lines	
	+-----+	+-----+	
	+-----+	+-----+	
	16-bit	74	
	Address	Opcodes	
	Bus		
	+-----+	+-----+	
	+-----+	+-----+	
	Serial	On-chip	
	I/O	Clock	
	+-----+	+-----+	
+-----+			

મેમરી ટ્રીક: "CHAIRS" - "ક્લોક, હાર્ડવેર ઇન્ટરપ્ટ, એડ્રેસ બસ, ઇન્સ્ટ્રક્શન સેટ, રજિસ્ટર્સ, સિરિયલ I/O"

પ્રશ્ન 3(અ) [3 ગુણ]

8051 ના કોઈપણ એક ટાઈમર મોડને સમજાવો.

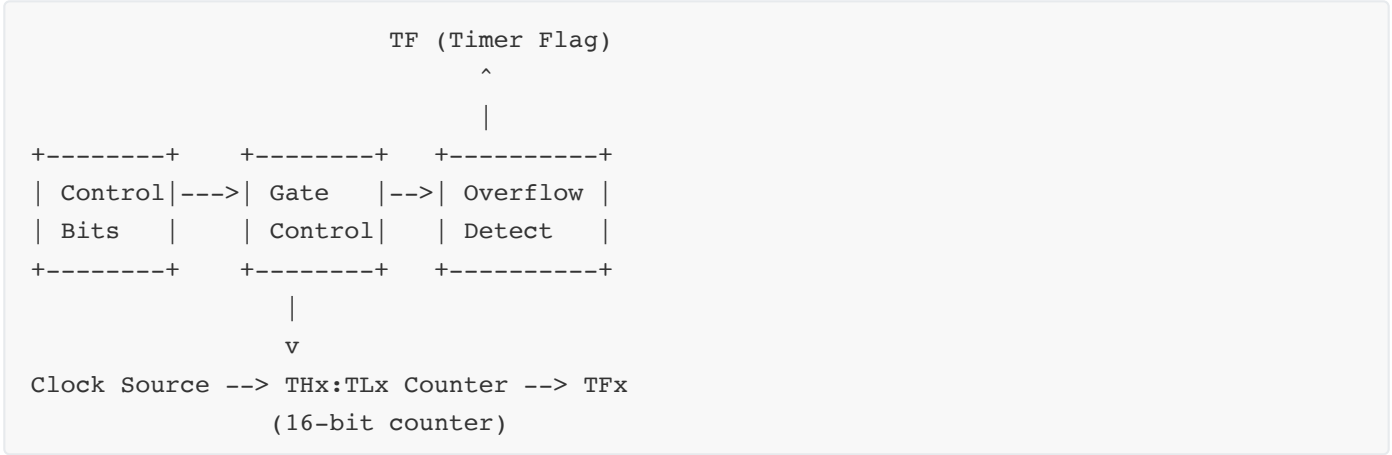
જવાબ:

મોડ 1: 16-બિટ ટાઈમર/કાઉન્ટર

ફીચર	વર્ણન
ટાઈમર સ્ટ્રક્ચર	THx અને TLx રજિસ્ટર્સ વાપરીને 16-બિટ ટાઈમર
ઓપરેશન	0000H થી FFFFH સુધી ગણતરી કરે છે, પછી TF ફ્લેગ સેટ કરે છે
કાઉન્ટર સાઈઝ	કુલ 16-બિટ કાઉન્ટર ( $2^{16} = 65,536$ કાઉન્ટ્સ)
રજિસ્ટર્સ	THx (હાઈ બાઈટ) અને TLx (લો બાઈટ)

સાચાગ્રામ:





મેમરી ટ્રીક: "MOGC" - "મોડ 1 ઓવરફ્લો ડિટેક્શન, ગેટ કંટ્રોલ, કમ્પ્લીટ 16-બિટ"

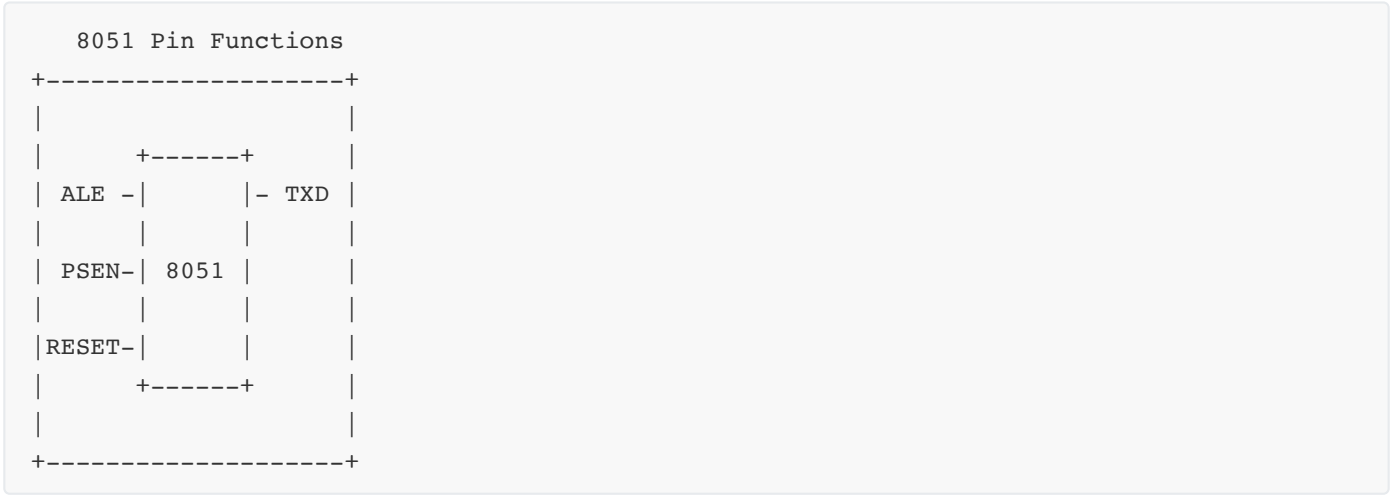
### પ્રશ્ન 3(બ) [4 ગુણ]

8051 માટે ALE, PSEN, RESET અને TXD પિનનું ફંક્શન લખો.

જવાબ:

પિન	ફંક્શન
ALE	એડ્રેસ લેય એનેબલ - પોર્ટ 0 માંથી એડ્રેસનો લો બાઈટ લેય કરવા માટે કંટ્રોલ સિગ્નલ પૂરું પાડે છે
PSEN	પ્રોગ્રામ સ્ટોર એનેબલ - બાહ્ય પ્રોગ્રામ મેમરી એક્સેસ માટે રીડ સ્ટ્રોબ
RESET	રીસેટ ઇનપુટ - 2 મશીન સાયકલ સુધી HIGH રાખવાથી CPU ને પ્રારંભિક સ્થિતિમાં ફોર્સ કરે છે
TXD	ટ્રાન્સમિટ ડેટા - સિરિયલ ડેટા ટ્રાન્સમિશન માટે સિરિયલ પોર્ટ આઉટપુટ પિન

સાચાગ્રામ:



મેમરી ટ્રીક: "APTR" - "એડ્રેસ લેય, પ્રોગ્રામ સ્ટોર, ટોટલ રીસેટ, ટ્રાન્સમિટ ડેટા"

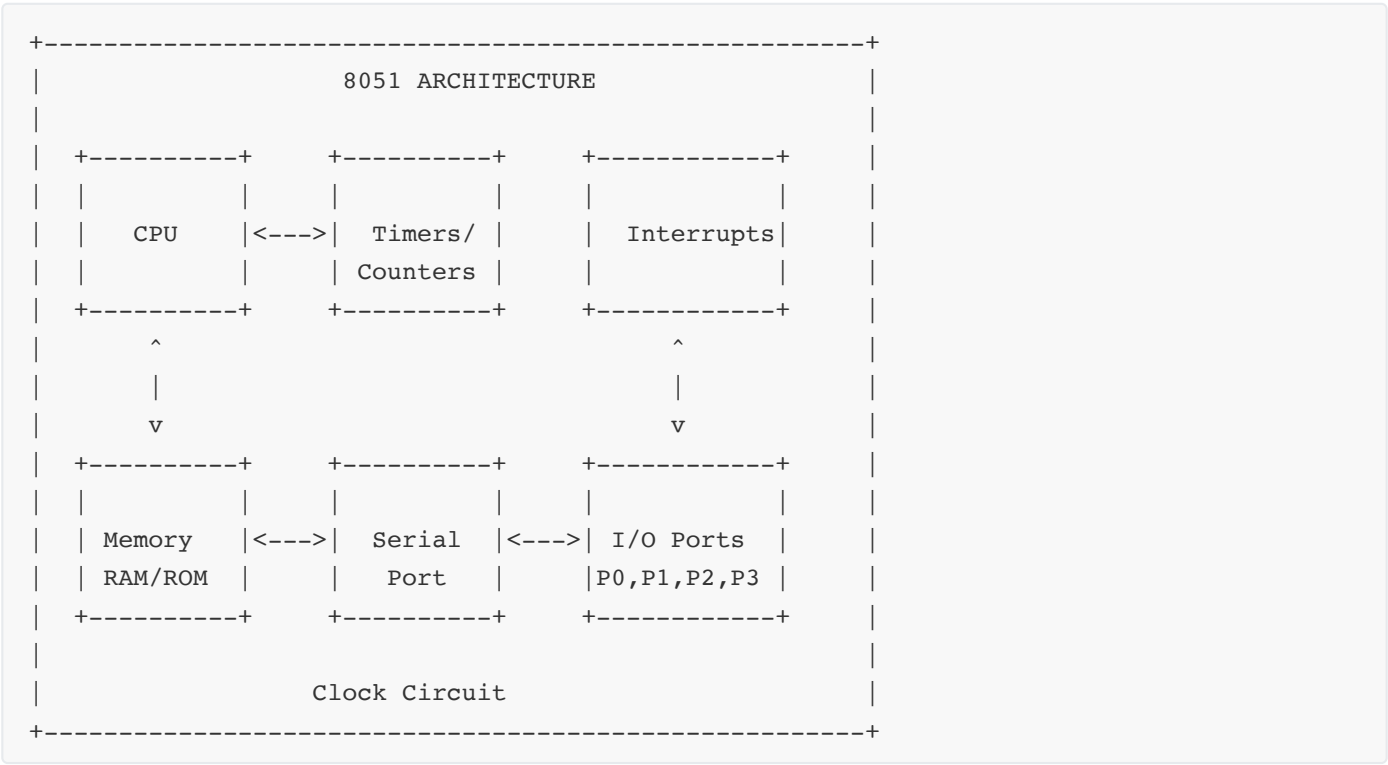
### પ્રશ્ન 3(ક) [7 ગુણ]

8051 માઇક્રોકંટ્રોલરના દરેક બ્લોકના કાર્યો સમજાવો.

જવાબ:

બ્લોક	કાર્ય
CPU	8-બિટ પ્રોસેસર જે ઇન્સ્ટ્રક્શન ફેચ અને એક્ઝિક્યુટ કરે છે
મેમરી	4KB ઇન્ટરનલ ROM અને 128 બાઈટ્સ ઇન્ટરનલ RAM
I/O પોર્ટ્સ	ચાર 8-બિટ બિડાયરેક્શનલ I/O પોર્ટ્સ (P0-P3)
ટાઈમર/કાઉન્ટર	ટાઈમિંગ અને કાઉન્ટિંગ માટે બે 16-બિટ ટાઈમર/કાઉન્ટર
સિરિયલ પોર્ટ	સિરિયલ કમ્યુનિકેશન માટે ફુલ-ડુપ્લેક્સ UART
ઇન્ટરપ્ટ	બે પ્રાયોરિટી લેવલ સાથે પાંચ ઇન્ટરપ્ટ સોર્સ
ક્લોક સર્કિટ	તમામ ઓપરેશન માટે ટાઈમિંગ પૂરું પાડે છે

સાચાગ્રામ:



મેમરી ટ્રીક: "CRIMSON" - "CPU, RAM/ROM, I/O, મેમરી, સિરિયલ પોર્ટ, ઓસિલેટર, ઇન્ટરપ્ટ"

### પ્રશ્ન 3(અ OR) [3 ગુણ]

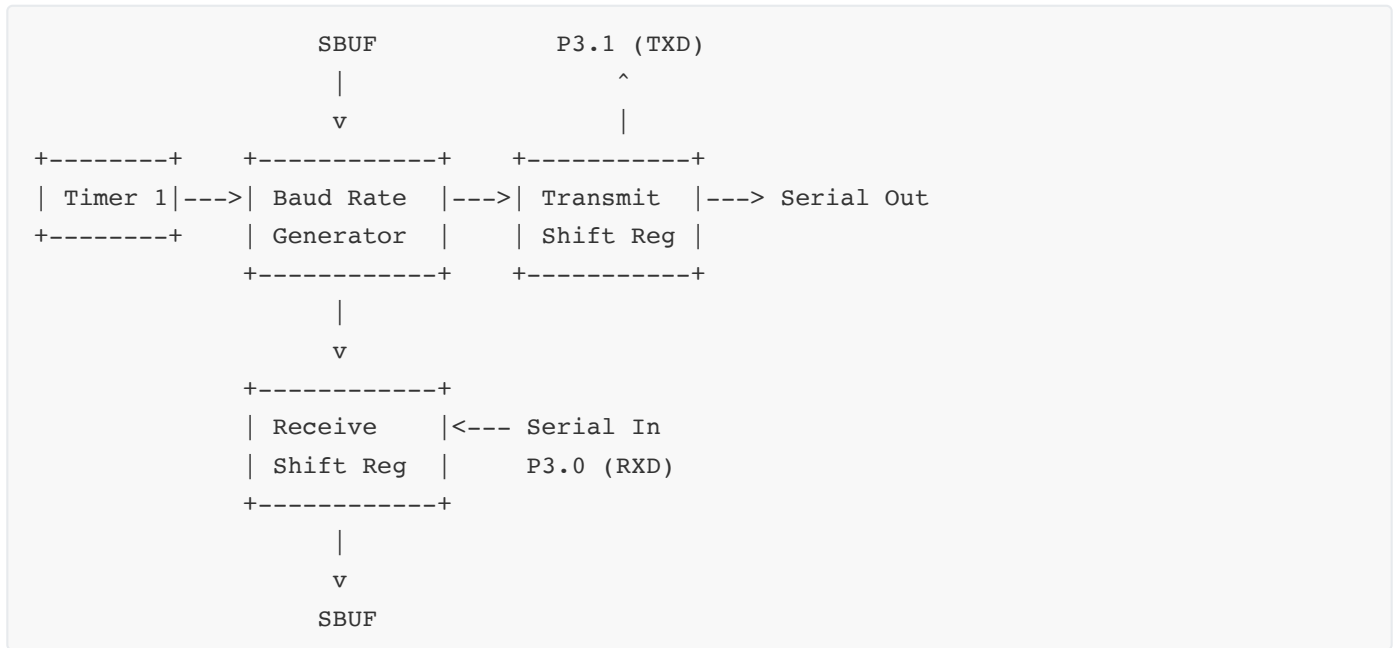
8051 ના કોઈપણ એક સીરિયલ કોમ્યુનિકેશન મોડને સમજાવો.

જવાબ:

મોડ 1: 8-બિટ UART

ફીચર	વર્ણન
ફોર્મેટ	10 બિટ્સ (સ્ટાર્ટ બિટ, 8 ડેટા બિટ્સ, સ્ટોપ બિટ)
બોડ રેટ	વેરિએબલ, ટાઈમર 1 દ્વારા નક્કી થાય છે
ડેટા ડાયરેક્શન	કુલ-ડુપ્લેક્સ (એક સાથે ટ્રાન્સમિટ અને રિસીવ)
પિન્સ	ટ્રાન્સમિટ માટે TXD (P3.1), રિસીવ માટે RXD (P3.0)

## સાચાગ્રામ:



**મેમરી ટ્રીક:** "FADS" - "ફોર્મેટ 10-બિટ, ઓટો બોડ ટાઈમર 1 થી, ડુપ્લેક્સ મોડ, સ્ટાન્ડર્ડ UART"

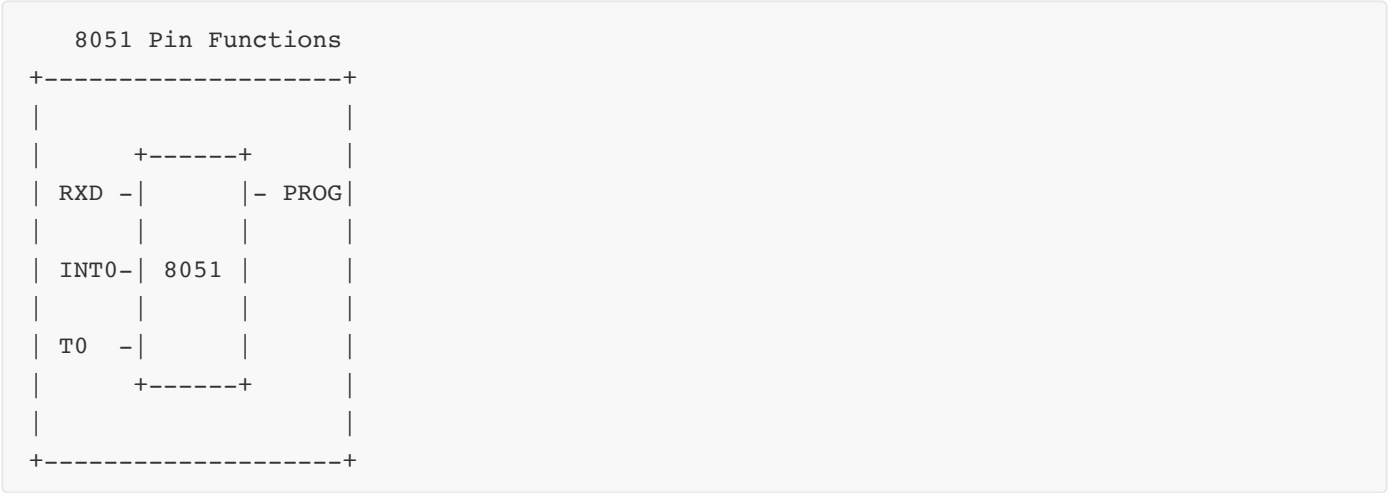
**પ્રશ્ન 3(બ OR) [4 ગુણ]**

**8051 માટે RXD, INT0, T0 અને PROG પિનનું ફંક્શન લખો.**

### જવાબ:

પિન	ફંક્શન
<b>RXD (P3.0)</b>	રિસીવ ડેટા - સિરિયલ ડેટા રિસેપ્શન માટે સિરિયલ પોર્ટ ઇનપુટ પિન
<b>INT0 (P3.2)</b>	એક્સટર્નલ ઇન્ટરપ્ટ 0 - બાહ્ય ઇન્ટરપ્ટ ટ્રિગર કરી શકે તેવો ઇનપુટ
<b>T0 (P3.4)</b>	ટાઈમર 0 - ટાઈમર/કાઉન્ટર 0 માટે બાહ્ય કાઉન્ટ ઇનપુટ
<b>PROG (EA)</b>	પ્રોગ્રામ એનેબલ - જ્યારે LOW હોય, ત્યારે CPU ને બાહ્ય મેમરીમાંથી કોડ ફેચ કરવા ફોર્સ કરે છે

## સાયાગ્રામ:



**મેમરી ટ્રીક:** "RIPE" - "રિસીવ ડેટા, ઇન્ટરપ્ટ ટ્રિગર, પલ્સ કાઉન્ટિંગ, એક્સટર્નલ મેમરી"

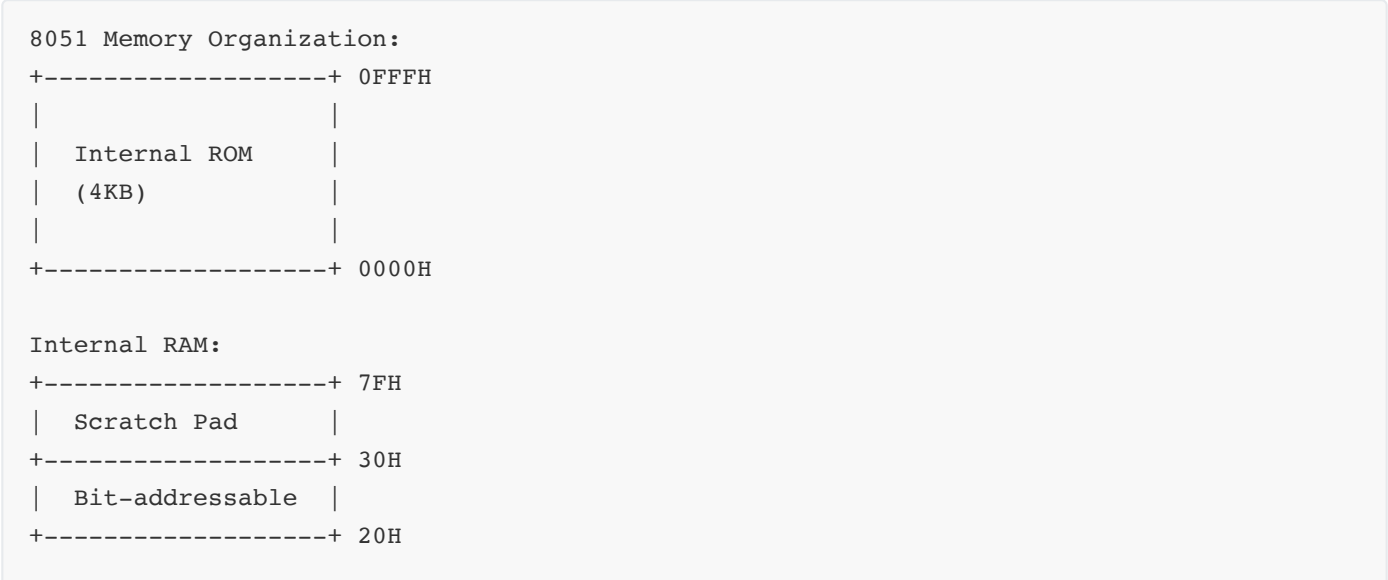
પ્રશ્ન 3(ક OR) [7 ગુણ]

8051 માટે ALU, PC, DPTR, RS0, RS1, આંતરિક RAM અને આંતરિક ROM નું વર્ણન કરો.

જવાબ:

ઘટક	વર્ણન
ALU	અર્થમેટિક લોજિક યુનિટ - ગાણિતિક અને લોજિકલ ઓપરેશન કરે છે
PC	પ્રોગ્રામ કાઉન્ટર - 16-બિટ રજિસ્ટર જે આગલી ઇન્સ્ટ્રક્શનને પોઇન્ટ કરે છે
DPTR	ડેટા પોઇન્ટર - 16-બિટ રજિસ્ટર (DPH+DPL) બાહ્ય મેમરી એડ્રેસિંગ માટે
RS0, RS1	PSW માં રજિસ્ટર બેંક સિલેક્ટ બિટ્સ - ચાર રજિસ્ટર બેંક્સમાંથી એક પસંદ કરે છે
આંતરિક RAM	128 બાઇટ્સ ઓન-ચિપ RAM (00H-7FH) વેરિએબલ્સ અને સ્ટેક માટે
આંતરિક ROM	4KB ઓન-ચિપ ROM (0000H-0FFFH) પ્રોગ્રામ સ્ટોરેજ માટે

સાચાગ્રામ:



```
| Register Banks |
| (RS0,RS1 select) |
+-----+ 00H
```

મેમરી ટ્રીક: "APRID" - "ALU પ્રોસેસ કરે, PC યાદ રાખે, રજિસ્ટર બેંક સિલેક્ટ, ઇન્ટરનલ મેમરી, DPTR પોઇન્ટ કરે"

## પ્રશ્ન 4(અ) [3 ગુણ]

08H ને 02H થી વિભાજિત કરવા માટે એસેમ્બલી લાખામાં પ્રોગ્રામ વિકસાવો.

જવાબ:

```
MOV A, #08H ; ડિવિડન્ડ 08H એક્યુમ્યુલેટરમાં લોડ કરો
MOV B, #02H ; ડિવાઇઝર 02H B રજિસ્ટરમાં લોડ કરો
DIV AB ; A ને B વડે ભાગો (A=ભાગફળ, B=શેષ)
MOV R0, A ; ભાગફળ R0 માં સ્ટોર કરો (04H)
MOV R1, B ; શેષ R1 માં સ્ટોર કરો (00H)
```

સાચાગ્રામ:

Before DIV AB:	After DIV AB:
+-----+	+-----+
A: 08H	A: 04H   (Quotient)
+-----+	+-----+
+-----+	+-----+
B: 02H	B: 00H   (Remainder)
+-----+	+-----+

મેમરી ટ્રીક: "LDDS" - "લોડ ડિવિડન્ડ, ડિવાઇઝર B માં, ડિવાઇડ, સ્ટોર રિઝલ્ટ"

## પ્રશ્ન 4(બ) [4 ગુણ]

76H અને 32H ઉમેરવા માટે એસેમ્બલી લાખામાં પ્રોગ્રામ વિકસાવો.

જવાબ:

```
MOV A, #76H ; પહેલો નંબર 76H એક્યુમ્યુલેટરમાં લોડ કરો
MOV R0, #32H ; બીજો નંબર 32H R0 માં લોડ કરો
ADD A, R0 ; R0 ને A માં ઉમેરો (76H + 32H = A8H)
MOV R1, A ; પરિણામ R1 માં સ્ટોર કરો (A8H)
JNC DONE ; જો કેરી ન આવે તો જમ્પ કરો
MOV R2, #01H ; જો કેરી આવે તો, R2 માં 1 સ્ટોર કરો
DONE: NOP ; પ્રોગ્રામ પૂરો કરો
```

સાચાગ્રામ:

+-----+
+-----+
+-----+
| 76H | + ? | 32H | = ? | A8H | + Carry Flag
+-----+
+-----+
+-----+

Calculation:

76H = 0111 0110

+ 32H = 0011 0010

-----

A8H = 1010 1000

મેમરી ટ્રીક: "LASER" - "લોડ A, સ્ટોર સેકન્ડ નંબર, એક્ઝિક્યુટ એડિશન, રિઝલ્ટ સ્ટોર"

## પ્રશ્ન 4(ક) [7 ગુણ]

એડ્રેસિંગ મોડ શું છે? તેને 8051 માટે વર્ગીકૃત કરો.

જવાબ:

એડ્રેસિંગ મોડ: ઇન્સ્ટ્રક્શન માટે ઓપરેન્ડ/ડેટાનું સ્થાન સ્પષ્ટ કરવાની પદ્ધતિ.

એડ્રેસિંગ મોડ	વર્ણન	ઉદાહરણ
રજિસ્ટર	ઓપરેન્ડ રજિસ્ટરમાં	MOV A, R0 (R0 ને A માં મુવ કરે)
ડાયરેક્ટ	ઓપરેન્ડ ચોક્કસ મેમરી લોકેશન પર	MOV A, 30H (30H પરથી ડેટા A માં મુવ કરે)
રજિસ્ટર ઇન્ડાયરેક્ટ	રજિસ્ટરમાં ઓપરેન્ડનું એડ્રેસ	MOV A, @R0 (R0 માં સ્ટોર એડ્રેસ પરથી ડેટા A માં મુવ કરે)
ઇમીડિયેટ	ઓપરેન્ડ ઇન્સ્ટ્રક્શનનો ભાગ છે	MOV A, #55H (A માં 55H લોડ કરે)
ઇન્ડેક્સ	બેઝ એડ્રેસ + ઓફસેટ	MOVC A, @A+DPTR (A+DPTR પર કોડ બાઈટ મેળવે)
બિટ	વ્યક્તિગત બિટ એડ્રેસેબલ	SETB P1.0 (પોર્ટ 1 ના બિટ 0 ને સેટ કરે)
ઇમ્પ્લાઈડ/ઇનહેરન્ટ	ઓપરેન્ડ ઇન્સ્ટ્રક્શન દ્વારા સૂચિત	RRC A (A ને કેરી સાથે જમણી બાજુ રોટેટ કરે)

ડાયાગ્રામ:

+-----+
+-----+
+-----+
Register		Direct		Indirect										
MOV A, R5		MOV A, 40H		MOV A, @R1										
+---+ +---+		+---+ +---+		+---+ +---+										
	A	<---	R5			A	<---	40H			A	<---	X	
+---+ +---+		+---+ +---+		+---+ +---+										
+-----+ +-----+														
		^												
	+---+---+													
		R1=X												
	+-----+													
+-----+

મેમરી ટ્રીક: "RIDDIB" - "રજિસ્ટર, ઇમીડિયેટ, ડાયરેક્ટ, ડેટા ઇન્ડાયરેક્ટ, ઇન્ડેક્સ, બિટ"

## પ્રશ્ન 4(અ OR) [3 ગુણ]

08H અને 02H નો ગુણાકાર કરવા માટે એસેમ્બલી લાખામાં પ્રોગ્રામ વિકસાવો.

જવાબ:

```

MOV A, #08H    ; પહેલો નંબર 08H એક્યુમ્યુલેટરમાં લોડ કરો
MOV B, #02H    ; બીજો નંબર 02H B રજિસ્ટરમાં લોડ કરો
MUL AB         ; A અને B નો ગુણાકાર કરો (B:A = પરિણામ)
MOV R0, A      ; લો-બાઈટ પરિણામ R0 માં સ્ટોર કરો (10H)
MOV R1, B      ; હાઈ-બાઈટ પરિણામ R1 માં સ્ટોર કરો (00H)

```

સાચાગ્રામ:

Before MUL AB:	After MUL AB:
+-----+	+-----+
A: 08H	A: 10H   (08H × 02H = 10H)
+-----+	+-----+
+-----+	+-----+
B: 02H	B: 00H   (High byte = 00H)
+-----+	+-----+

મેમરી ટ્રીક: "LMSR" - "લોડ નંબર્સ, મલ્ટિપ્લાય, સ્ટોર રિઝલ્ટ"

## પ્રશ્ન 4(બ) [4 ગુણ]

76H માંથી 32H બાદ કરવા માટે એસેમ્બલી લાખામાં પ્રોગ્રામ વિકસાવો.

જવાબ:

```

MOV A, #32H    ; 32H એક્યુમ્યુલેટરમાં લોડ કરો
MOV R0, #76H   ; 76H R0 માં લોડ કરો
CLR C          ; કેરી ફ્લેગ ક્લિયર કરો (બોરો ફ્લેગ)
SUBB A, R0      ; A માંથી R0 બોરો સાથે બાદ કરો (32H - 76H = BCH)
MOV R1, A       ; પરિણામ R1 માં સ્ટોર કરો (BCH, જે -44H દર્શાવે છે)

```

સાચાગ્રામ:

+-----+	+-----+	+-----+
32H	- ?	76H   = ?   BCH   (represents -44H)
+-----+	+-----+	+-----+

Calculation:

```

32H = 0011 0010
- 76H = 0111 0110
-----
BCH = 1011 1100 (two's complement of 44H)

```

મેમરી ટ્રીક: "LESS" - "લોડ ફર્સ્ટ નંબર, એનેબલ બોરો (CLR C), સબટ્રેક્ટ, સ્ટોર"

## પ્રશ્ન 4(ક) [7 ગુણ]

Instruction set ના પ્રકારોની સૂચિ બનાવો. કોઈપણ ત્રણને એક ઉદાહરણ સાથે સમજાવો.

જવાબ:

ઇન્સ્ટ્રક્શન ગ્રુપ	વર્ણન	ઉદાહરણ
અર્થમેટિક	ગાણિતિક ઓપરેશન	ADD A, R0 (R0 ને A માં ઉમેરો)
લોજિકલ	લોજિકલ ઓપરેશન	ANL A, #0FH (A ને 0FH સાથે AND કરો)
ડેટા ટ્રાન્સફર	લોકેશન વચ્ચે ડેટા ખસેડો	MOV A, R7 (R7 ને A માં ખસેડો)
બ્રાન્ચ	પ્રોગ્રામ ફ્લો બદલો	JNZ LOOP (જો A શૂન્ય ન હોય તો જમ્પ કરો)
બિટ મેનિપ્યુલેશન	વ્યક્તિગત બિટ પર ઓપરેશન	SETB P1.0 (પોર્ટ 1 ના બિટ 0 ને સેટ કરો)
મશીન કંટ્રોલ	પ્રોસેસર ઓપરેશન કંટ્રોલ	NOP (કોઈ ઓપરેશન નહીં)

સમજાવેલા ઇન્સ્ટ્રક્શન્સ:

### 1. ડેટા ટ્રાન્સફર ઇન્સ્ટ્રક્શન્સ:

- રજિસ્ટર્સ, મેમરી, અથવા I/O પોર્ટ્સ વચ્ચે ડેટા ખસેડે છે
- ઉદાહરણ: MOV A, 30H - મેમરી લોકેશન 30H માંથી એક્યુમ્યુલેટરમાં ડેટા ખસેડે છે
- ઓપરેશન:  $A \leftarrow [30H]$

### 2. અર્થમેટિક ઇન્સ્ટ્રક્શન્સ:

- ઉમેરવું, બાદ કરવું વગેરે જેવા ગાણિતિક ઓપરેશન કરે છે
- ઉદાહરણ: ADD A, R0 - R0 ની સામગ્રી એક્યુમ્યુલેટરમાં ઉમેરે છે
- ઓપરેશન:  $A \leftarrow A + R0$

### 3. લોજિકલ ઇન્સ્ટ્રક્શન્સ:

- AND, OR, XOR, NOT જેવા લોજિકલ ઓપરેશન કરે છે
- ઉદાહરણ: ANL A, #0FH - અપર નિબલ માસ્ક કરે છે (માત્ર લોઅર નિબલ રાખે છે)
- ઓપરેશન:  $A \leftarrow A \text{ AND } 0FH$

સાચાગ્રામ:

8051 Instruction Types			
+-----+-----+			
	+-----+	+-----+	
	Data	Branch	
	Transfer	Instructions	
	+-----+	+-----+	
	+-----+	+-----+	
	Arithmetic	Bit	



		Instructions		Manipulation	
		+-----+		+-----+	
		+-----+		+-----+	
		Logical		Machine	
		Instructions		Control	
		+-----+		+-----+	
		+-----+		+-----+	

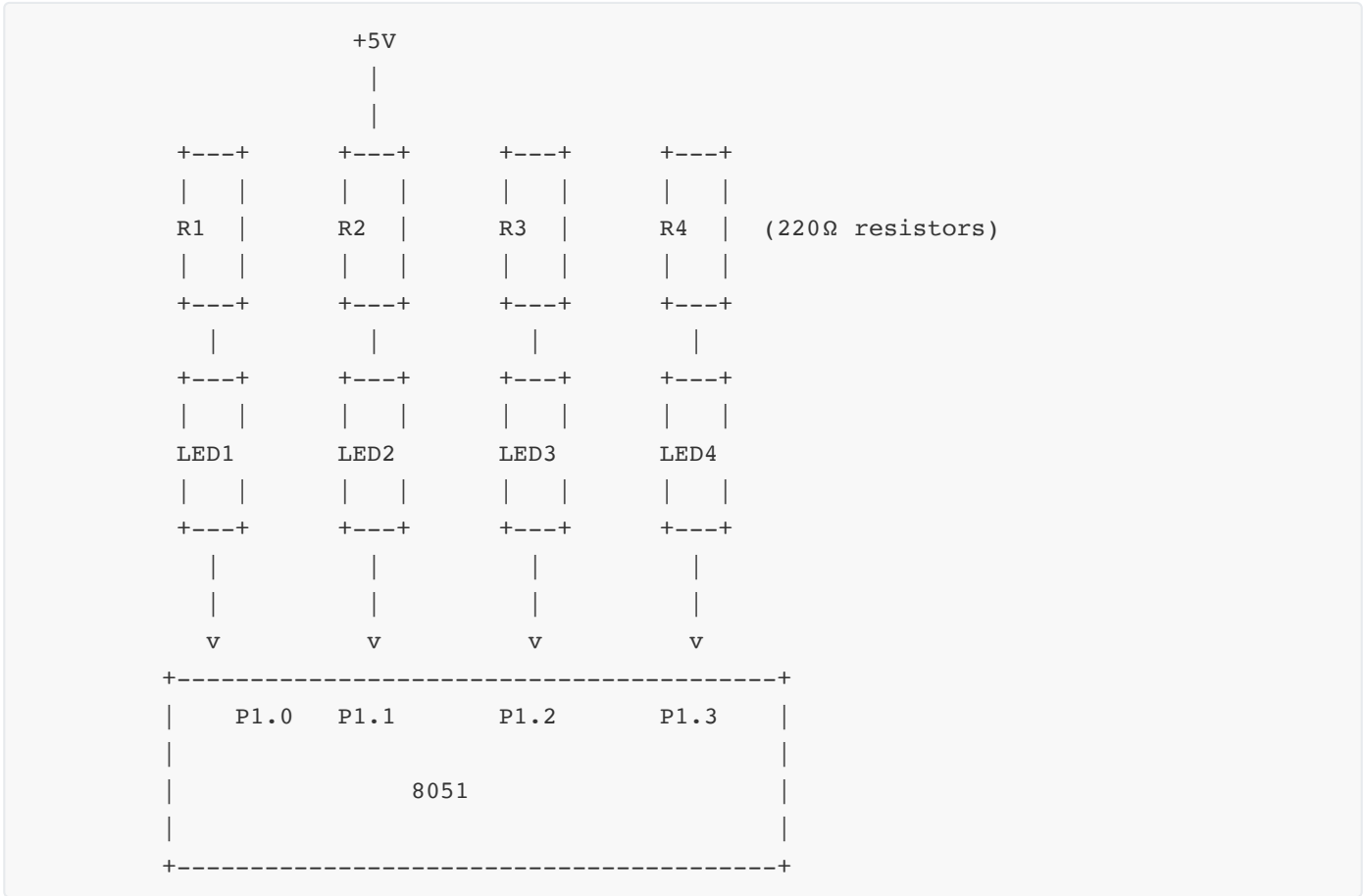
મેમરી ટ્રીક: "BALDM" - "બ્રાન્ચ, અર્થમેટિક, લોજિકલ, ડેટા ટ્રાન્સફર, મશીન કંટ્રોલ"

## પ્રશ્ન 5(અ) [3 ગુણ]

8051 માઇક્રોકન્ટ્રોલર સાથે ચાર એલઇડીનું ઇન્ટરફેસિંગ દોરો.

જવાબ:

ડાયાગ્રામ:



જરૂરી ઘટકો:

- 8051 માઇક્રોકન્ટ્રોલર
- ચાર LED
- ચાર કરંટ લિમિટિંગ રેસિસ્ટર (220Ω)
- પાવર સપ્લાય

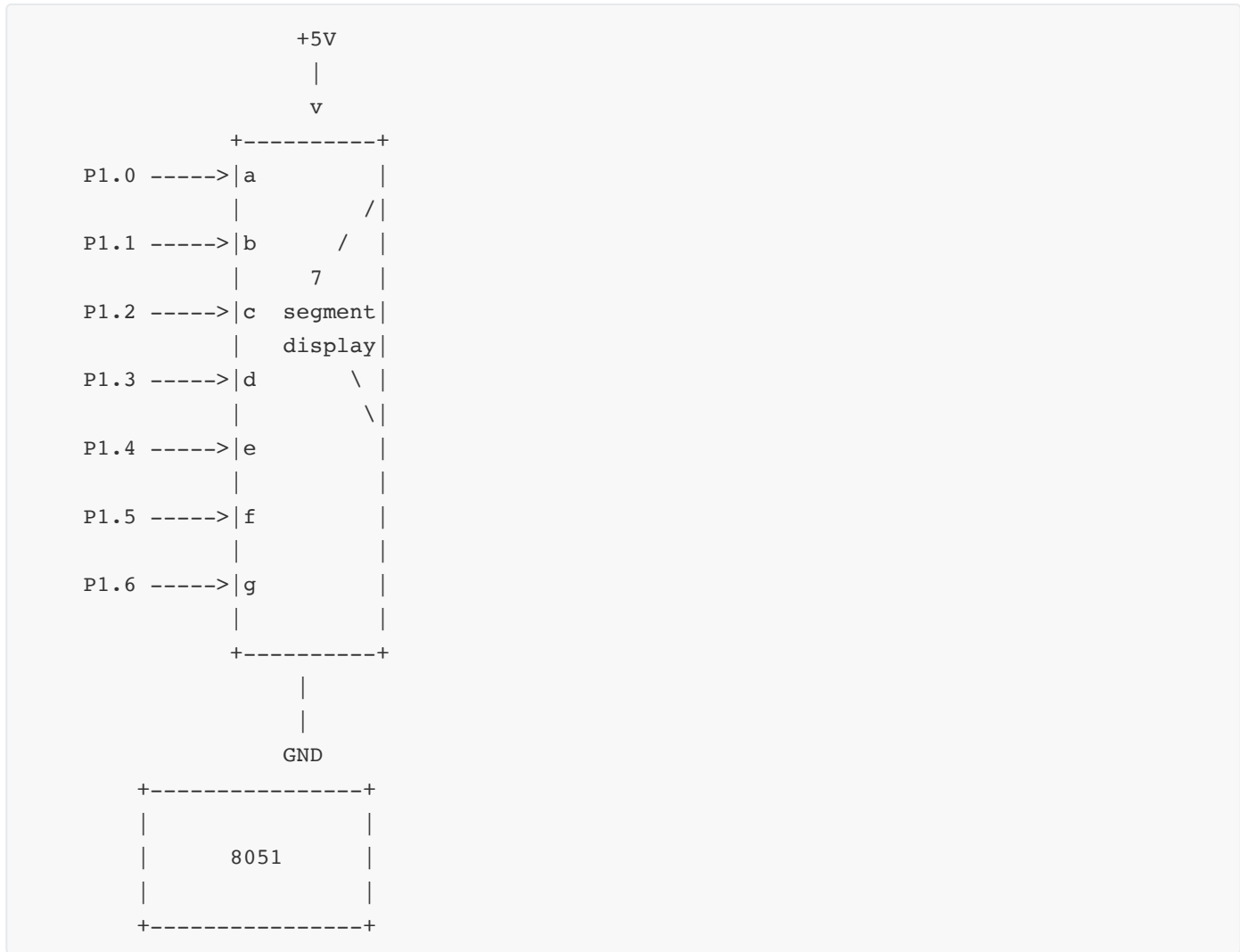
મેમરી ટ્રીક: "PALS" - "પોર્ટ પિન, એક્ટિવ-લો કંટ્રોલ, LED, સિમ્પલ સર્કિટ"

## પ્રશ્ન 5(બ) [4 ગુણ]

8051 માઇક્રોકન્ટ્રોલર સાથે 7 સેગમેન્ટ એલઇડીનું ઇન્ટરફેસિંગ દોરો.

જવાબ:

ડાયાગ્રામ:



; 0-9 અંકો માટે સેગમેન્ટ પેટર્ન ડિફાઇન કરો

**DIGITS:** DB 3FH, 06H, 5BH, 4FH, 66H, 6DH, 7DH, 07H, 7FH, 6FH

; અંક 5 ડિસ્પ્લે કરો

MOV A, #6DH ; 5 માટે સેગમેન્ટ પેટર્ન

MOV P1, A ; પોર્ટ P1 પર મોકલો

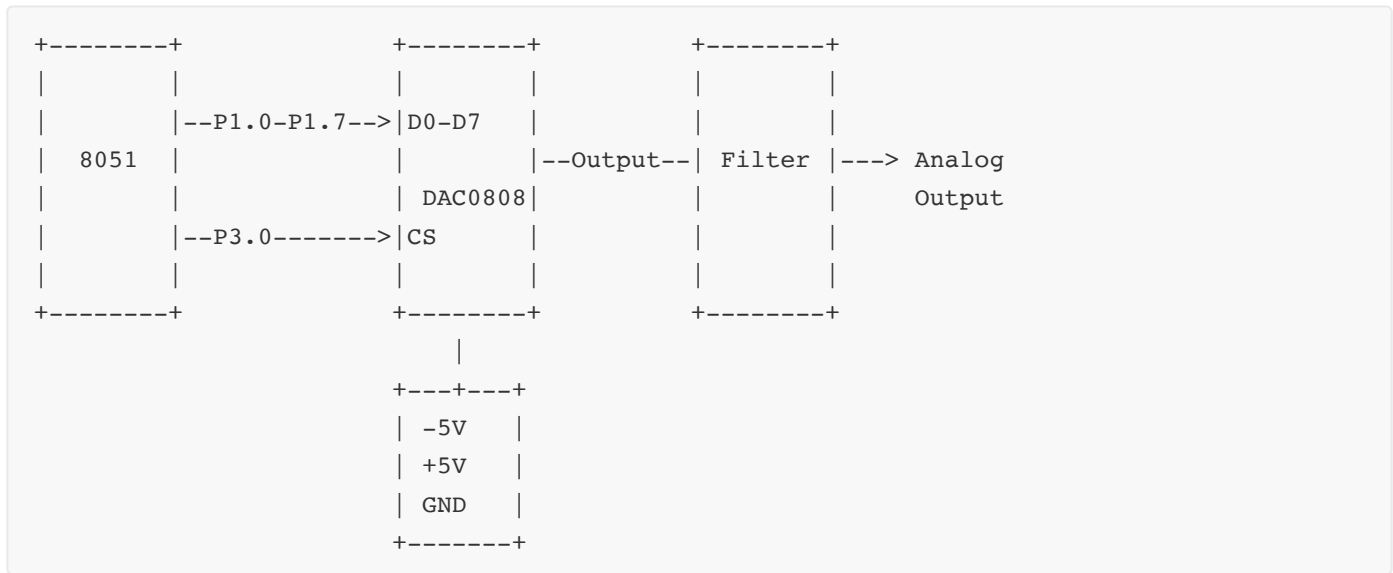
મેમરી ટ્રીક: "SPACE-7" - "સેવન પિન્સ, પેટર્ન સેગમેન્ટ, અર્થિંગ કોમન, ઇઝી ડિસ્પ્લે"

## પ્રશ્ન 5(ક) [7 ગુણ]

8051 માઇક્રોકન્ટ્રોલર સાથે DAC નું ઇન્ટરફેસિંગ સમજાવો અને જરૂરી પ્રોગ્રામ લખો.

જવાબ:

સાચાગ્રામ:



જરૂરી ઘટકો:

- 8051 માઇક્રોકન્ટ્રોલર
- DAC0808 (8-બિટ ડિજિટલ-ટુ-એનાલોગ કન્વર્ટર)
- આઉટપુટ બફરિંગ માટે ઓપરેશનલ એમ્પ્લિફાયર
- સ્મુથિંગ માટે RC ફિલ્ટર
- પાવર સપ્લાય

કનેક્શન્સ:

- P1.0-P1.7 → D0-D7 (8-બિટ ડિજિટલ ઇનપુટ)
- P3.0 → CS (ચિપ સિલેક્ટ)

સોટૂથ વેવ જનરેટ કરવા માટે પ્રોગ્રામ:

```

START:  MOV R0, #00H      ; R0 ને 0 થી ઇનિશિયલાઇઝ કરો
LOOP:   MOV P1, R0        ; DAC પર વેલ્યુ આઉટપુટ કરો
        CALL DELAY        ; થોડો સમય રાહ જુઓ
        INC R0            ; વેલ્યુ વધારો
        SJMP LOOP         ; સોટૂથ વેવ બનાવવા રિપીટ કરો

DELAY:  MOV R7, #50       ; ડિલે કાઉન્ટર લોડ કરો
DELAY1: MOV R6, #255      ; ઇનર લૂપ કાઉન્ટર
DELAY2: DJNZ R6, DELAY2   ; R6 ને ઝીરો થાય ત્યાં સુધી ઘટાડો
        DJNZ R7, DELAY1   ; R7 ને ઝીરો થાય ત્યાં સુધી ઘટાડો
        RET              ; સબરૂટિનથી પાછા ફરો

```

#### કાર્યપ્રણાલી:

1. ડિજિટલ વેલ્યુ પોર્ટ 1 પર આઉટપુટ કરવામાં આવે છે
2. DAC 8-બિટ ડિજિટલ વેલ્યુને પ્રપોર્શનલ એનાલોગ વોલ્ટેજમાં કન્વર્ટ કરે છે
3. ફિલ્ટર આઉટપુટ સિગ્નલને સ્મૂથ કરે છે
4. પ્રોગ્રામ આઉટપુટ વેલ્યુને વધારીને સોટૂથ વેવ બનાવે છે

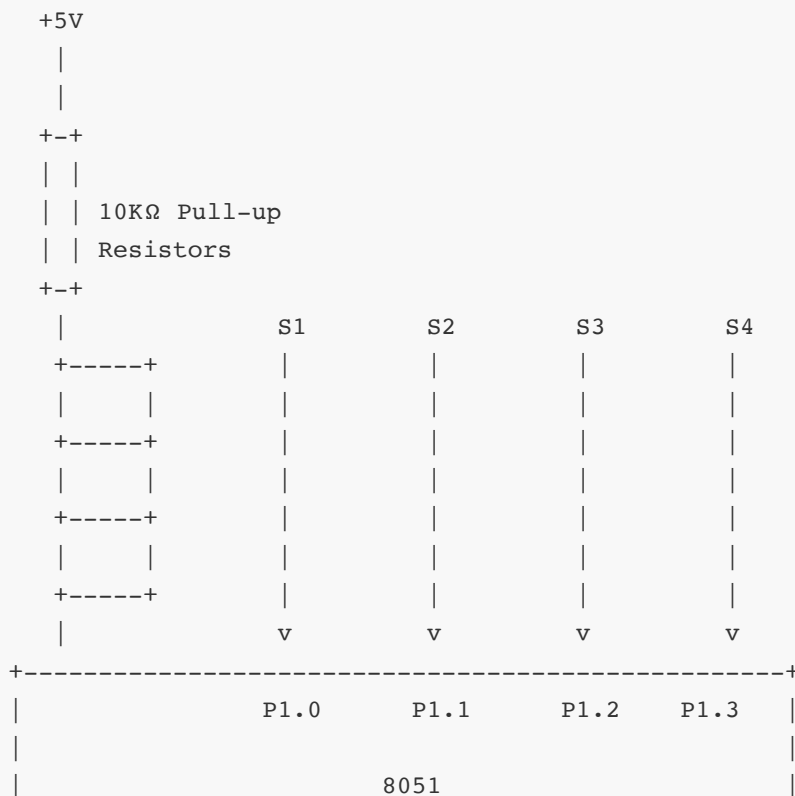
મેમરી ટ્રીક: "DICAF" - "ડિજિટલ ઇનપુટ, ઇન્ક્રિમેન્ટ, કન્વર્ટ ટુ એનાલોગ, એમ્પ્લિફાય, ફિલ્ટર"

## પ્રશ્ન 5(અ OR) [3 ગુણ]

8051 માઇક્રોકન્ટ્રોલર સાથે ચાર સ્વીચોનું ઇન્ટરફેસિંગ દોરો.

જવાબ:

સાચાગ્રામ:



|  
+-----+  
|

**જરૂરી ઘટકો:**

- 8051 માઇક્રોકન્ટ્રોલર
- ચાર પુશ બટન (નોર્મલી ઓપન)
- પુલ-અપ રેસિસ્ટર્સ (10KΩ)
- પાવર સપ્લાય

**કાર્યપ્રણાલી:**

- સ્વિચ દબાવવા પર ગ્રાઉન્ડ સાથે જોડાય છે
- સ્વિચ ઓપન હોય ત્યારે પોર્ટ પિન HIGH (1) વાંચે છે
- સ્વિચ દબાવેલ હોય ત્યારે પોર્ટ પિન LOW (0) વાંચે છે

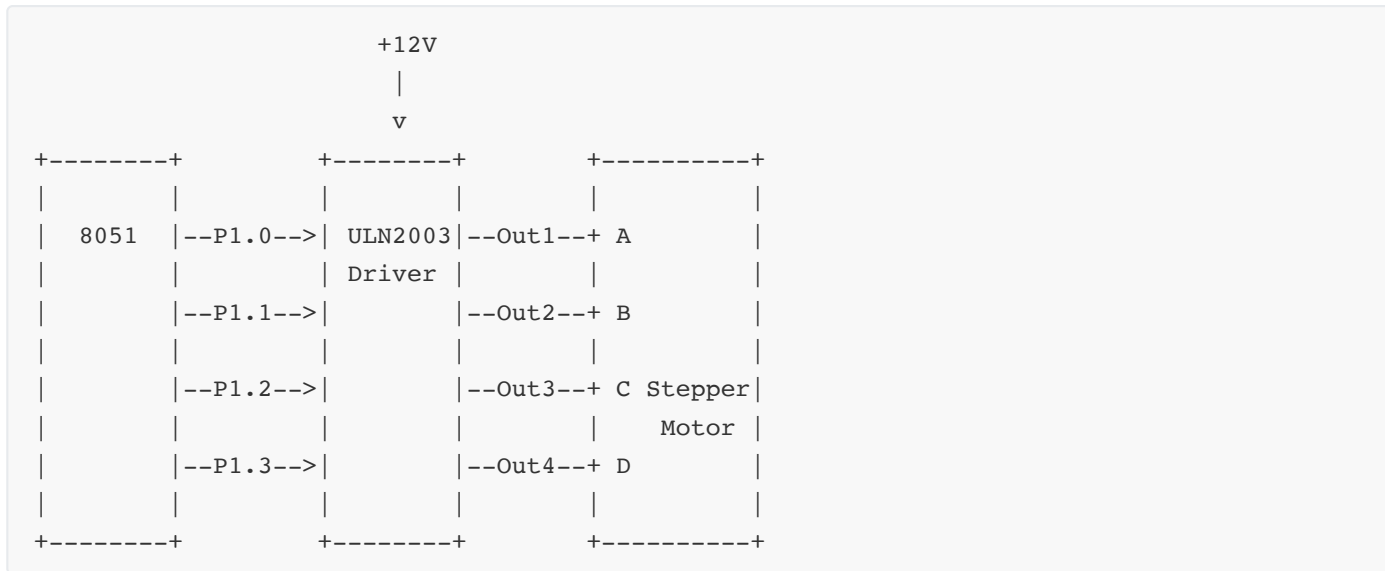
**મેમરી ટ્રીક:** "PIPS" - "પુલ-અપ્સ, ઇનપુટ પિન્સ, પ્રેસ ફોર ઝીરો, સ્વિચિસ"

**પ્રશ્ન 5(બ) [4 ગુણ]**

8051 માઇક્રોકન્ટ્રોલર સાથે સ્ટેપર મોટરનું ઇન્ટરફેસિંગ દોરો.

**જવાબ:**

**ડાયાગ્રામ:**

**જરૂરી ઘટકો:**

- 8051 માઇક્રોકન્ટ્રોલર
- ULN2003 ડ્રાઇવર IC
- સ્ટેપર મોટર (4-ફેઝ)
- પાવર સપ્લાય

**એક્સાઇટેશન સિક્વન્સ:**

સ્ટેપ	P1.3 (D)	P1.2 (C)	P1.1 (B)	P1.0 (A)	હેક્સ વેલ્યુ
1	0	0	0	1	01H
2	0	0	1	0	02H
3	0	1	0	0	04H
4	1	0	0	0	08H

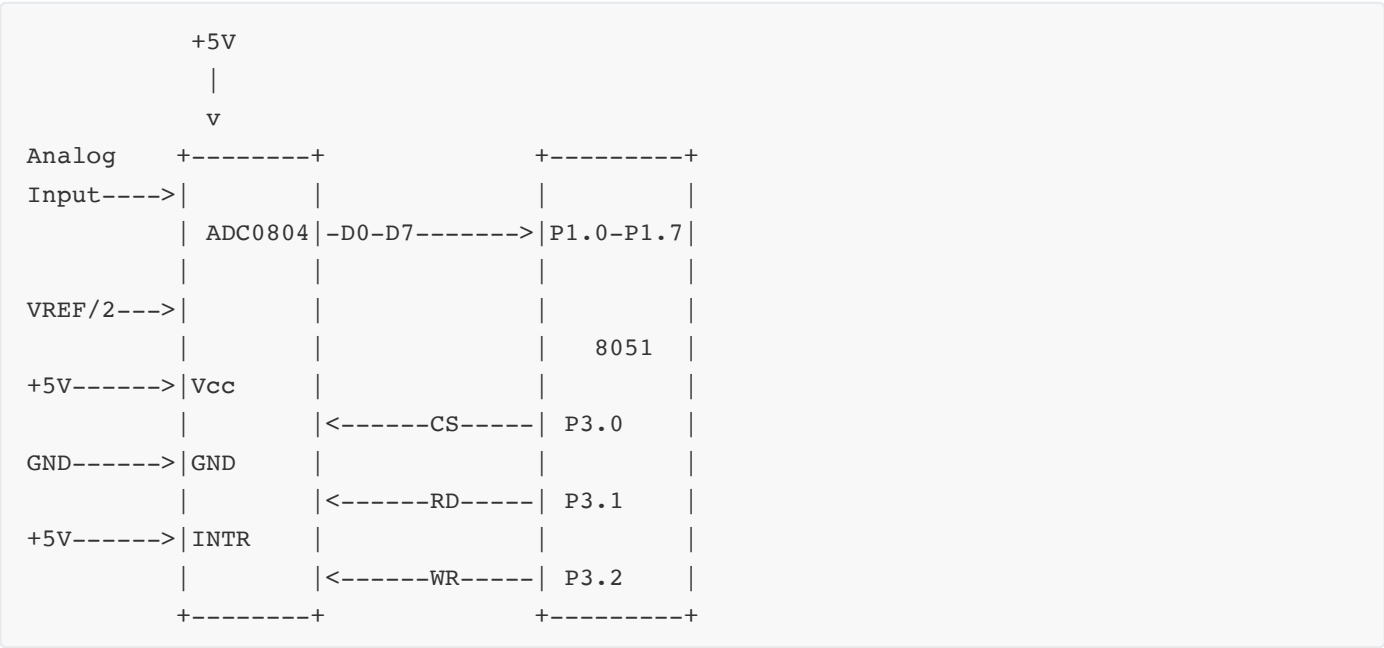
મેમરી ટ્રીક: "CUPS" - "કંટ્રોલર આઉટપુટ સિક્વન્સ, ULN2003 એમ્પ્લિફાય, ફેઝ એનર્જાઇઝ, સ્ટેપિંગ મોશન"

પ્રશ્ન 5(ક) [7 ગુણ]

8051 માઇક્રોકન્ટ્રોલર સાથે ADC નું ઇન્ટરફેસિંગ સમજાવો અને જરૂરી પ્રોગ્રામ લખો.

જવાબ:

ડાયાગ્રામ:



જરૂરી ઘટકો:

- 8051 માઇક્રોકન્ટ્રોલર
- ADC0804 (8-બિટ એનાલોગ-ટુ-ડિજિટલ કન્વર્ટર)
- રેફરન્સ વોલ્ટેજ સોર્સ
- ઇનપુટ કન્ડિશનિંગ સર્કિટ (નથી બતાવેલ)

કનેક્શન:

- P1.0-P1.7 ← D0-D7 (ADC માંથી 8-બિટ ડિજિટલ આઉટપુટ)
- P3.0 → CS (ચિપ સિલેક્ટ)
- P3.1 → RD (રીડ)

- P3.2 → WR (રાઈટ)

### એનાલોગ ઇનપુટ વાંચવા માટે પ્રોગ્રામ:

```

START:  MOV P1, #0FFH      ; P1 ને ઇનપુટ પોર્ટ તરીકે કન્ફિગર કરો

READ:   CLR P3.0           ; ADC એનેબલ કરો (CS = 0)
          CLR P3.2           ; કન્વર્ઝન શરૂ કરો (WR = 0)
          NOP                ; નાનો ડિલે
          NOP
          SETB P3.2          ; WR = 1

WAIT:   JB P3.3, WAIT      ; કન્વર્ઝન માટે રાહ જુઓ (INTR = 0)

          CLR P3.1           ; ડેટા વાંચવા માટે RD = 0
          MOV A, P1           ; કન્વર્ટ કરેલી વેલ્યુ વાંચો
          SETB P3.1          ; RD = 1
          SETB P3.0          ; ADC ડિસેબલ કરો (CS = 1)

PROCESS:                ; જરૂરિયાત મુજબ ડેટા પ્રોસેસ કરો
          ; ઉદાહરણ: R0 માં સ્ટોર કરો
          MOV R0, A

          SJMP READ          ; સતત કન્વર્ઝન માટે રિપીટ કરો
  
```

### કાર્યપ્રણાલી:

1. કંટ્રોલર સ્ટાર્ટ કન્વર્ઝન સિગ્નલ મોકલે છે
2. ADC એનાલોગ ઇનપુટને 8-બિટ ડિજિટલ વેલ્યુમાં કન્વર્ટ કરે છે
3. કંટ્રોલર કન્વર્ઝન પૂર્ણ થયા પછી ડિજિટલ વેલ્યુ વાંચે છે
4. પ્રોગ્રામ જરૂરિયાત મુજબ ડિજિટલ વેલ્યુને પ્રોસેસ કરે છે

**મેમરી ટ્રીક:** "CARSW" - "કન્વર્ટ એનાલોગ, રીડ ડિજિટલ, સ્ટાર્ટ કન્વર્ઝન, વેઇટ ફોર કમ્પ્લીશન"