

પ્રશ્ન 1(અ) [3 ગુણ]

RISC અને CISC ની સરખામણી કરો.

જવાબ:

લક્ષણ	RISC	CISC
સૂચનાઓ	સરળ, નિશ્ચિત લંબાઈ	જટિલ, અલગ-અલગ લંબાઈ
અમલીકરણ	સિંગલ સાયકલ	મલ્ટીપલ સાયકલ
એડ્રેસિંગ મોડ	ઓછા	ઘણા
રજિસ્ટર્સ	વધારે	ઓછા
ડિઝાઇન ફોકસ	હાર્ડવેર સરળતા	કોડ ડેન્સિટી

યાદ રાખવા માટે: "RISC સરળતાથી સૂચનાઓ પૂર્ણ કરે છે"

પ્રશ્ન 1(બ) [4 ગુણ]

વોન-ન્યુમેન અને હાર્વર્ડ આર્કિટેક્ચરની તુલના કરો.

જવાબ:

લક્ષણ	વોન-ન્યુમેન	હાર્વર્ડ
મેમરી	એક શોર્ડ મેમરી	અલગ પ્રોગ્રામ અને ડેટા મેમરી
બસ	ડેટા અને સૂચનાઓ માટે એક બસ	અલગ બસ
સ્પીડ	ધીમી (મેમરી બોટલનેક)	ઝડપી (પેરેલલ એક્સેસ)
જટિલતા	સરળ ડિઝાઇન	વધુ જટિલ
ઉપયોગ	જનરલ કમ્પ્યુટિંગ	રીયલ-ટાઇમ સિસ્ટમ

ડાયાગ્રામ:

Von-Neumann :

```

+-----+           +-----+
| CPU    | <=====> | Memory |
+-----+           +-----+

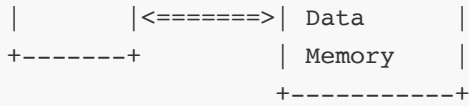
```

Harvard :

```

+-----+           +-----+
| CPU    | =====> | Program  |
|         |           | Memory   |
|         |           +-----+
|         |           +-----+

```



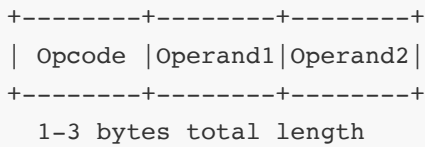
યાદ રાખવા માટે: "હાર્વર્ડ પાસે અલગ જગ્યાઓ છે"

પ્રશ્ન 1(ક) [7 ગુણ]

સમજાવો: 8085 ઈન્સ્ટ્રક્શન ફોર્મેટ, કંટ્રોલ યુનિટ, મશીન સાયકલ, ALU

જવાબ:

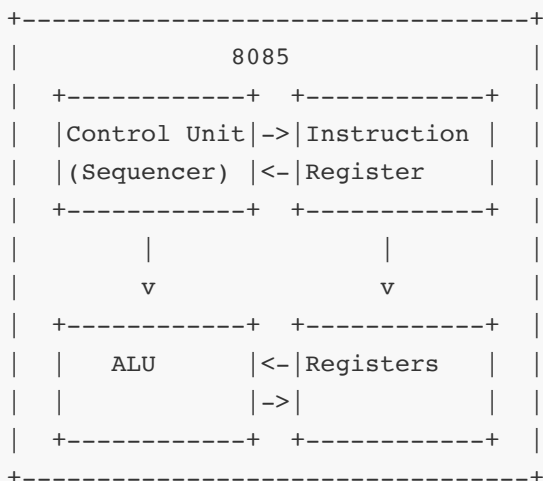
ઈન્સ્ટ્રક્શન ફોર્મેટ:



કમ્પોનન્ટ	કાર્ય
ઈન્સ્ટ્રક્શન ફોર્મેટ	1-3 બાઈટ સ્ટ્રક્ચર ઓપકોડ અને ઓપરેન્ડ સાથે
કંટ્રોલ યુનિટ	સૂચનાઓ ફેચ અને ડિકોડ કરે; સિગ્નલ પેદા કરે
મશીન સાયકલ	મૂળભૂત ઓપરેશન સાયકલ (T-સ્ટેટ્સ)
ALU	ગાણિતિક અને લોજિકલ ઓપરેશન કરે

- ઈન્સ્ટ્રક્શન ફોર્મેટ: ઓપકોડ (3-8 બિટ્સ) અને 0-2 ઓપરેન્ડ્સ ધરાવે છે
- કંટ્રોલ યુનિટ: પ્રોસેસરનું હૃદય જે બધા ઓપરેશન્સનું સંચાલન કરે છે
- મશીન સાયકલ: ફેચ, ડિકોડ, એક્ઝિક્યુટ ફેઝ ધરાવે છે
- ALU: ડેટા પર ADD/SUB/AND/OR/XOR ઓપરેશન કરે છે

સાચાગ્રામ:



યાદ રાખવા માટે: "CIMA: કંટ્રોલ સમજે, મશીન ક્રિયા કરે"

પ્રશ્ન 1(ક OR) [7 ગુણ]

માઇક્રોપ્રોસેસર અને માઇક્રોકંટ્રોલરની સરખામણી કરો.

જવાબ:

લક્ષણ	માઇક્રોપ્રોસેસર	માઇક્રોકંટ્રોલર
ડિઝાઇન	માત્ર CPU	CPU + પેરિફેરલ્સ
મેમરી	બાહ્ય	આંતરિક (RAM/ROM)
I/O પોર્ટ્સ	મર્યાદિત	બિલ્ટ-ઇન ઘણા
કિંમત	વધારે	ઓછી
ઉપયોગ	જનરલ કમ્પ્યુટિંગ	એમ્બેડેડ સિસ્ટમ
પાવર ખપત	વધારે	ઓછો
ઉદાહરણ	Intel 8085/8086	Intel 8051

ડાયાગ્રામ:

Microprocessor System:

```

+-----+ +-----+ +-----+
| CPU   |<-->| Memory|<-->| I/O   |
+-----+ +-----+ +-----+
        Separate chips needed
  
```

Microcontroller:

```

+-----+
| +-----+ +-----+ |
| | CPU   | | Memory| |
| +-----+ +-----+ |
|         |         | |
|         v         v |
| +-----+ +-----+ |
| | I/O, Timers, etc. | |
| +-----+ +-----+ |
+-----+
        All in one chip
  
```

યાદ રાખવા માટે: "માઇક્રો-P પ્રોસેસ કરે, માઇક્રો-C કંટ્રોલ કરે"

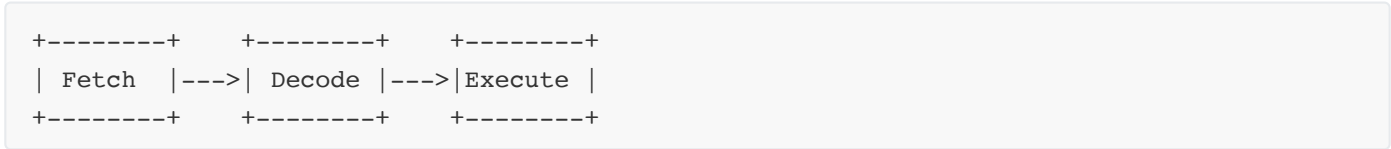
પ્રશ્ન 2(અ) [3 ગુણ]

માઇક્રોપ્રોસેસરમાં ઇન્સ્ટ્રક્શન ફેચિંગ, ડીકોડિંગ અને એક્ઝેક્યુશન ઓપરેશન સમજાવો.

જવાબ:

ફેઝ	ઓપરેશન
ફેચિંગ	CPU PC નો ઉપયોગ કરી મેમરીમાંથી સૂચના મેળવે
ડીકોડિંગ	ઓપરેશન પ્રકાર અને ઓપરેન્ડ નક્કી કરે
એક્ઝેક્યુશન	ખરેખર ઓપરેશન કરે

ડાયાગ્રામ:



યાદ રાખવા માટે: "FDE: પહેલા લે, પછી સમજે, અંતે કરે"

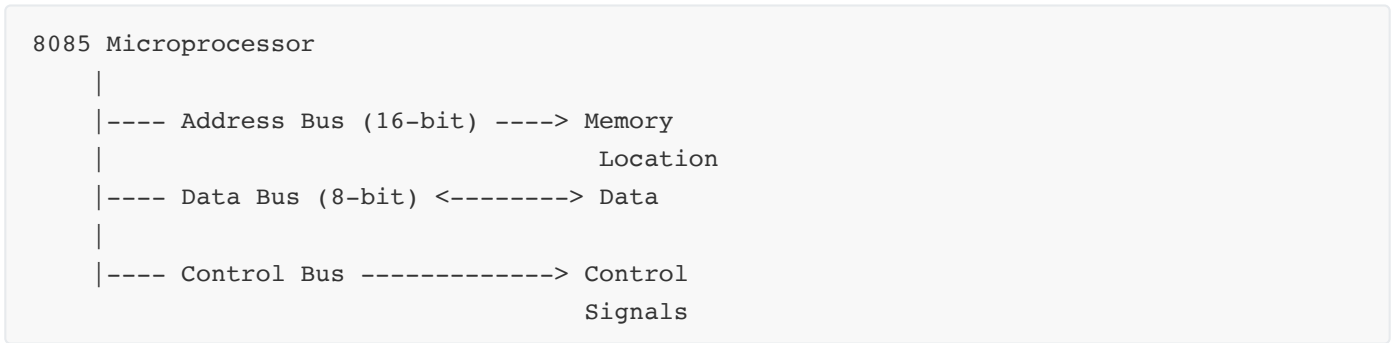
પ્રશ્ન 2(બ) [4 ગુણ]

8085 માઇક્રોપ્રોસેસરનું બસ ઓર્ગેનાઇઝેશન સમજાવો.

જવાબ:

બસ પ્રકાર	પહોળાઈ	કાર્ય
એડ્રેસ બસ	16-બિટ	મેમરી એડ્રેસ ટ્રાન્સફર કરે (A0-A15)
ડેટા બસ	8-બિટ	ડેટા ટ્રાન્સફર કરે (D0-D7)
કંટ્રોલ બસ	વિવિધ લાઇન્સ	ડેટા ફ્લો મેનેજ કરે (RD, WR, IO/M)
મલ્ટિપ્લેક્સ	AD0-AD7	લોઅર એડ્રેસ બિટ્સ + ડેટા બિટ્સ

ડાયાગ્રામ:



યાદ રાખવા માટે: "ADC: એડ્રેસ બતાવે, ડેટા વહે, કંટ્રોલ દિશા આપે"

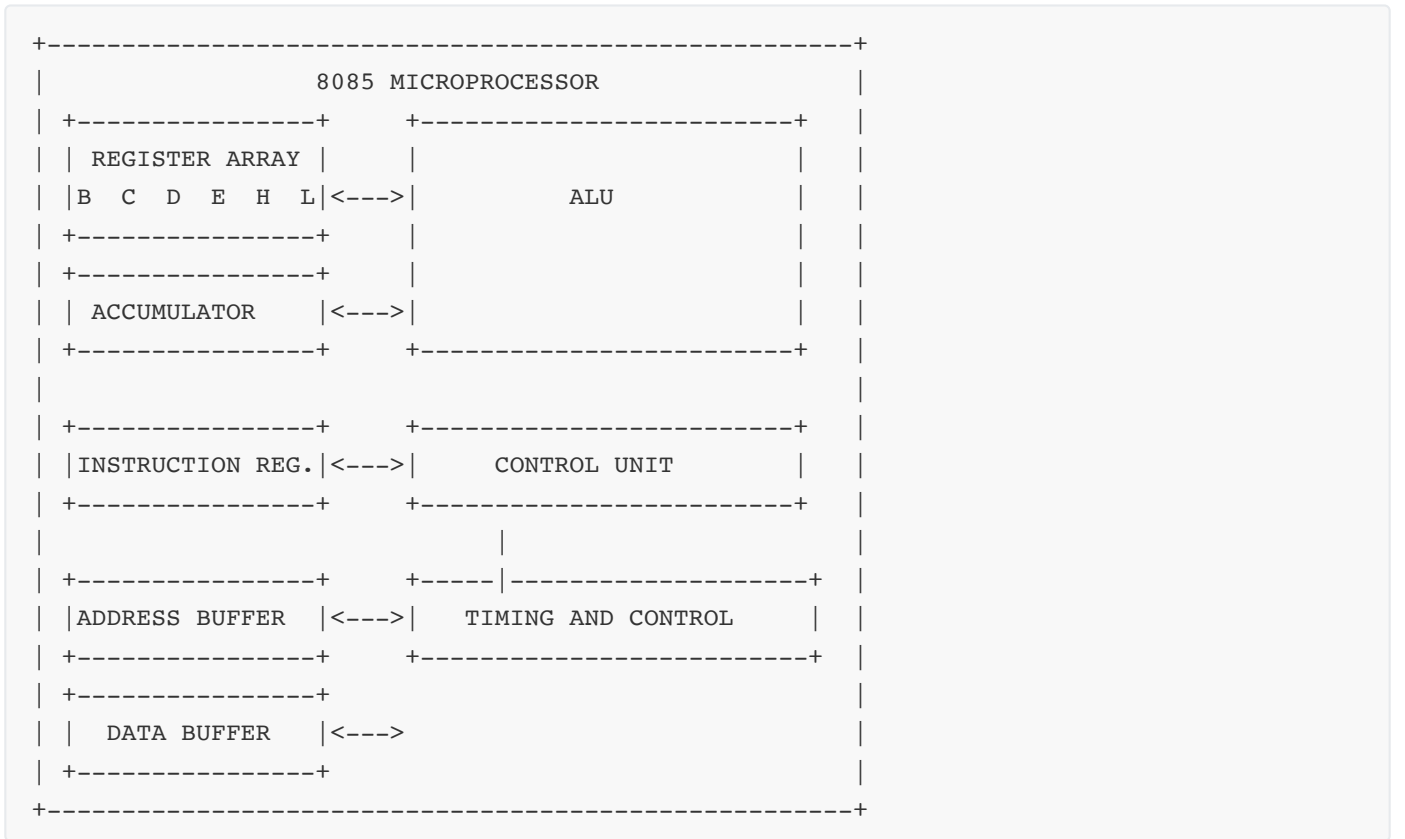
પ્રશ્ન 2(ક) [7 ગુણ]

આકૃતિની મદદથી 8085 માઇક્રોપ્રોસેસરના આર્કિટેક્ચરનું વર્ણન કરો.

જવાબ:

કમ્પોનન્ટ	કાર્ય
ALU	ગાણિતિક અને લોજિકલ ઓપરેશન્સ
રજિસ્ટર એરે	અસ્થાયી ડેટા સ્ટોરેજ (B,C,D,E,H,L)
એક્યુમ્યુલેટર	ગાણિતિક માટે મુખ્ય રજિસ્ટર
કંટ્રોલ યુનિટ	સૂચના કંટ્રોલ અને ટાઇમિંગ
ઈન્સ્ટ્રક્શન રજિસ્ટર	વર્તમાન સૂચના ધરાવે
ટાઇમિંગ & કંટ્રોલ	ટાઇમિંગ સિગ્નલ્સ જનરેટ કરે
એડ્રેસ બફર	એડ્રેસ બસ મેનેજ કરે
ડેટા બફર	ડેટા બસ ટ્રાન્સફર મેનેજ કરે

સાચાગ્રામ:



- **ALU:** ગાણિતિક અને લોજિકલ ઓપરેશન્સ કરે છે
- **કંટ્રોલ યુનિટ:** સૂચનાઓને ફેચ અને ડિકોડ કરે છે
- **રજિસ્ટર્સ:** પ્રોસેસિંગ દરમિયાન ડેટા અસ્થાયી રૂપે સ્ટોર કરે છે
- **બસેસ:** એડ્રેસ, ડેટા અને કંટ્રોલ સિગ્નલ્સ ટ્રાન્સફર કરે છે

યાદ રાખવા માટે: "ARCB: આર્કિટેક્ચર રજિસ્ટર કંટ્રોલ બસ ડેટા"

પ્રશ્ન 2(અ OR) [3 ગુણ]

8085 માઇક્રોપ્રોસેસર માટે એડ્રેસ અને ડેટા બસોનું ડી-મલ્ટીપ્લેક્સીંગ સમજાવો.

જવાબ:

સ્ટેપ	ક્રિયા
1	ALE સિગ્નલ હાઈ થાય
2	AD0-AD7 પર લોઅર એડ્રેસ (A0-A7) દેખાય
3	લેચ ALE નો ઉપયોગ કરી એડ્રેસ પકડે
4	ALE લો થાય, AD0-AD7 હવે ડેટા ટ્રાન્સફર કરે

ડાયાગ્રામ:

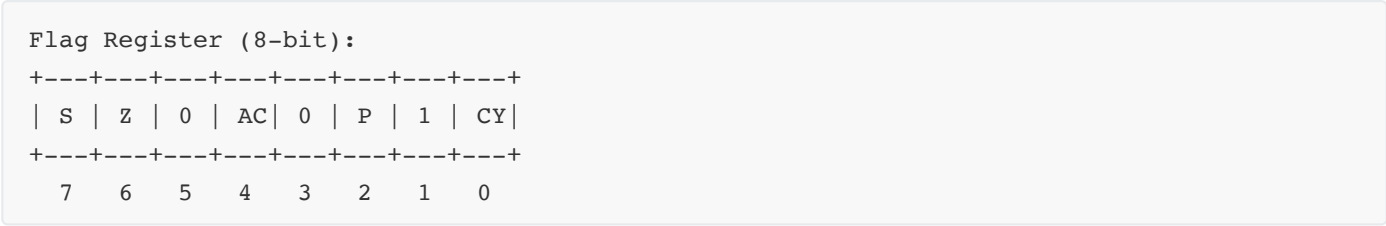


યાદ રાખવા માટે: "ALAD: ALE ડેટા પહેલા એડ્રેસ લેચ કરે"

પ્રશ્ન 2(બ OR) [4 ગુણ]

8085 માઇક્રોપ્રોસેસરનું ફ્લેગ રજિસ્ટર દોરો અને તેને સમજાવો.

જવાબ:



ફ્લેગ	નામ	સેટ થાય ત્યારે
S	સાઈન	પરિણામના બિટ 7 માં 1 હોય (નેગેટિવ)
Z	ઝીરો	પરિણામ શૂન્ય છે
AC	ઓકિઝલરી કેરી	બિટ 3 થી બિટ 4 માં કેરી આવે
P	પેરિટી	પરિણામમાં '1' ની સંખ્યા એવન (બેકી) હોય
CY	કેરી	બિટ 7 માંથી કેરી જનરેટ થાય

યાદ રાખવા માટે: "સુઝી ACની પરફેક્ટ કેરી"

પ્રશ્ન 2(ક OR) [7 ગુણ]

આકૃતિની મદદથી 8085 માઇક્રોપ્રોસેસરના પિન ડાયાગ્રામનું વર્ણન કરો.

જવાબ:

પિન ગ્રુપ	કાર્ય
એડ્રેસ/ડેટા	મલ્ટિપ્લેક્સડ AD0-AD7, A8-A15
કંટ્રોલ	RD, WR, IO/M, S0, S1, ALE, CLK
ઇન્ટરપ્ટ	INTR, RST 5.5-7.5, TRAP
DMA	HOLD, HLDA
પાવર	Vcc, Vss
સીરિયલ I/O	SID, SOD
રીસેટ	RESET IN, RESET OUT

સાચાગ્રામ:

		+-----+
	X1 --> 1	40 <-- Vcc
	X2 --> 2	39 <-- HOLD
RESET	OUT--> 3	38 <-- HLDA
RESET	IN --> 4	37 <-- CLK
IO/M	--> 5	36 <-- RESET IN
S1	--> 6	35 <-- READY
RD	--> 7	34 <-- IO/M
WR	--> 8	33 <-- S1
ALE	--> 9	32 <-- RD
S0	--> 10	31 <-- WR
A15	--> 11	30 <-- ALE
A14	--> 12	29 <-- S0
A13	--> 13	28 <-- A15
A12	--> 14	27 <-- A14
A11	--> 15	26 <-- A13
A10	--> 16	25 <-- A12
A9	--> 17	24 <-- A11
A8	--> 18	23 <-- A10
AD7	--> 19	22 <-- A9
AD6	--> 20	21 <-- A8
		+-----+

- **એડ્રેસ/ડેટા પિન્સ:** મલ્ટિપ્લેક્સડ પિન્સ ભૌતિક પિન બનાવે છે
- **કંટ્રોલ પિન્સ:** મેમરી અને I/O ઓપરેશન્સ કોઓર્ડિનેટ કરે છે
- **ઇન્ટરપ્ટ પિન્સ:** બાહ્ય ડિવાઇસને ઇન્ટરપ્ટ કરવા દે છે

- સીરિયલ પિન્સ: બેઝિક સીરિયલ કમ્યુનિકેશન પૂરું પાડે છે

યાદ રાખવા માટે: "ACID-PS: એક્સ-કંટ્રોલ-ઇન્ટરપ્ટ-DMA-પાવર-સીરિયલ"

પ્રશ્ન 3(અ) [3 ગુણ]

સ્ટેક, સ્ટેક પોઇન્ટર અને સ્ટેક ઓપરેશન સમજાવો.

જવાબ:

શબ્દ	વર્ણન
સ્ટેક	LIFO મેમરી એરિયા અસ્થાયી ડેટા સ્ટોરેજ માટે
સ્ટેક પોઇન્ટર	16-બિટ રજિસ્ટર જે સ્ટેક ટોપને પોઇન્ટ કરે
ઓપરેશન્સ	PUSH (સ્ટોર), POP (રીટ્રીવ)

ડાયાગ્રામ:

Memory:	Stack Pointer:
+-----+	+-----+
<-----	SP
+-----+	+-----+
Data	
+-----+	PUSH: SP--, M[SP]=data
Data	POP: data=M[SP], SP++
+-----+	

યાદ રાખવા માટે: "SP LIFO લેનને પોઇન્ટ કરે છે"

પ્રશ્ન 3(બ) [4 ગુણ]

8051 માઇક્રોકન્ટ્રોલરનો પિન ડાયાગ્રામ દોરો.

જવાબ:

8051 Microcontroller			
+-----+			
P1.0--	1	40	--VCC
P1.1--	2	39	--P0.0/AD0
P1.2--	3	38	--P0.1/AD1
P1.3--	4	37	--P0.2/AD2
P1.4--	5	36	--P0.3/AD3
P1.5--	6	35	--P0.4/AD4
P1.6--	7	34	--P0.5/AD5
P1.7--	8	33	--P0.6/AD6
RST --	9	32	--P0.7/AD7
P3.0/RXD	10	31	--EA/VPP
P3.1/TXD	11	30	--ALE/PROG
P3.2/INT0	12	29	--PSEN

P3.3/INT1	13	28	--P2.7/A15
P3.4/T0	14	27	--P2.6/A14
P3.5/T1	15	26	--P2.5/A13
P3.6/WR	16	25	--P2.4/A12
P3.7/RD	17	24	--P2.3/A11
XTAL2	-- 18	23	--P2.2/A10
XTAL1	-- 19	22	--P2.1/A9
VSS	-- 20	21	--P2.0/A8
+-----+			

પિન ગ્રુપ	કાર્ય
P0	પોર્ટ 0, એડ્રેસ/ડેટા સાથે મલ્ટિપ્લેક્સ
P1	પોર્ટ 1, જનરલ પર્પઝ I/O
P2	પોર્ટ 2, અપર એડ્રેસ અને I/O
P3	પોર્ટ 3, સ્પેશિયલ ફંક્શન્સ અને I/O

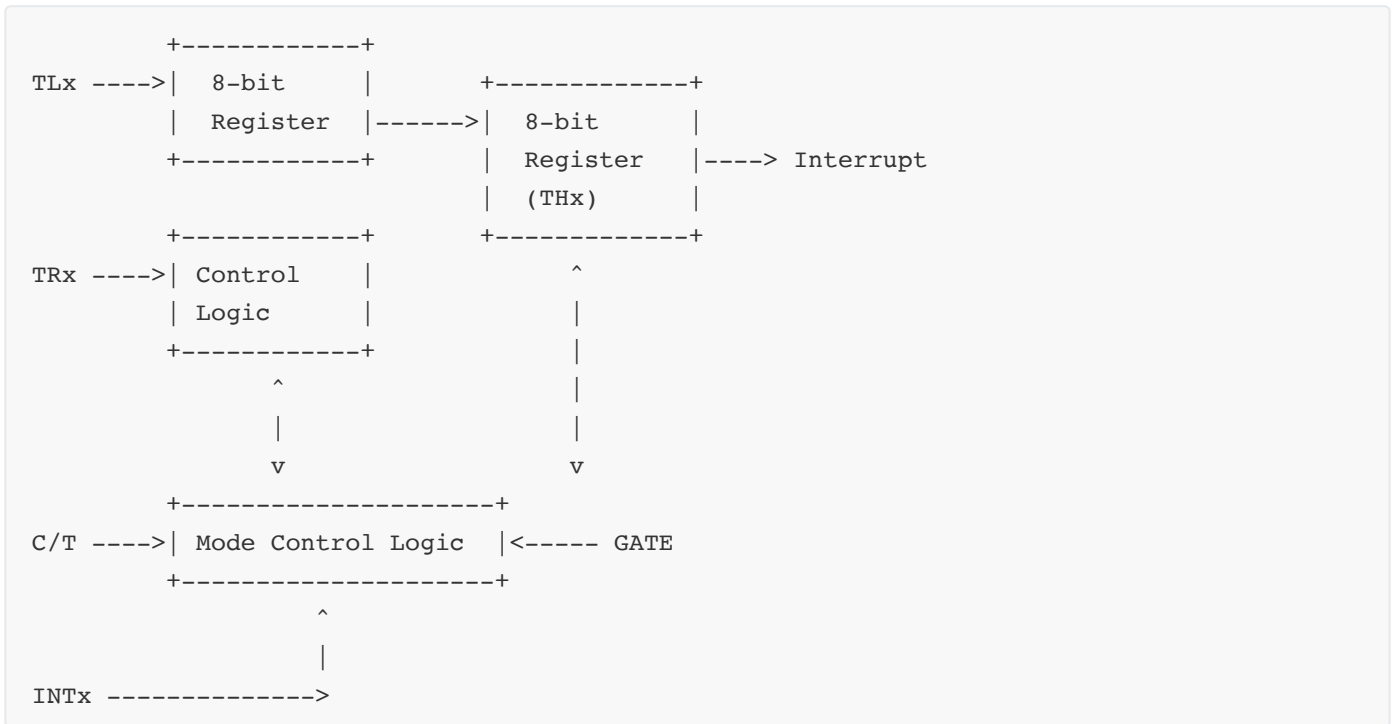
યાદ રાખવા માટે: "PORT 0123: ડેટા-જનરલ-એડ્રેસ-સ્પેશિયલ"

પ્રશ્ન 3(ક) [7 ગુણ] (ચાલુ)

8051 માઇક્રોકંટ્રોલરનો ટાઇમર્સ/કાઉન્ટર્સ લોજિક ડાયાગ્રામ દોરો અને વિવિધ મોડમાં તેની કામગીરી સમજાવો.

જવાબ:

Timer/Counter Diagram:



મોડ	ઓપરેશન
મોડ 0	13-બિટ ટાઇમર (5-બિટ TL, 8-બિટ TH)
મોડ 1	16-બિટ ટાઇમર (8-બિટ TL, 8-બિટ TH)
મોડ 2	8-બિટ ઓટો-રિલોડ (TL કાઉન્ટ, TH રીલોડ)
મોડ 3	સ્પ્લિટ ટાઇમર (માત્ર ટાઇમર 0)

- **ટાઇમર:** આંતરિક ક્લોક વાપરે, મશીન સાયકલ ગણે
- **કાઉન્ટર:** બાહ્ય ઇનપુટ વાપરે, બાહ્ય ઘટનાઓ ગણે
- **કંટ્રોલ બિટ્સ:** TMOD રજિસ્ટર મોડ સેટ કરે, TCON ઓપરેશન કંટ્રોલ કરે
- **મોડ્સ:** વિવિધ ટાઇમિંગ જરૂરિયાતો માટે અલગ-અલગ કોન્ફિગરેશન

યાદ રાખવા માટે: "MARC: મોડ ઓટો-રિલોડ કાઉન્ટ"

પ્રશ્ન 3(અ OR) [3 ગુણ]

માઇક્રોકંટ્રોલર્સનાં કોમન ફીચર્સની સૂચિ બનાવો.

જવાબ:

ફીચર	હેતુ
CPU કોર	સૂચનાઓ પ્રોસેસ કરવા
મેમરી (RAM/ROM)	પ્રોગ્રામ અને ડેટા સ્ટોર કરવા
I/O પોર્ટ્સ	બાહ્ય ડિવાઇસ સાથે ઇન્ટરફેસ
ટાઇમર/કાઉન્ટર	સમય અંતરાલ માપવા
ઇન્ટરપ્ટ	અસિંક્રોનસ ઘટનાઓ સંભાળવા
સીરિયલ કમ્યુનિકેશન	અન્ય ડિવાઇસ સાથે ડેટા ટ્રાન્સફર

યાદ રાખવા માટે: "CPU-TIS: CPU-RAM-I/O-ટાઇમર-ઇન્ટરપ્ટ-સીરિયલ"

પ્રશ્ન 3(બ OR) [4 ગુણ]

8051 માઇક્રોકંટ્રોલરનું ઇન્ટરનલ રેમ ઓર્ગેનાઇઝેશન સમજાવો.

જવાબ:

RAM એરિયા	એડ્રેસ રેન્જ	ઉપયોગ
રજિસ્ટર બેન્ક્સ	00H-1FH	R0-R7 (4 બેન્ક્સ)
બિટ-એડ્રેસેબલ	20H-2FH	128 બિટ્સ (0-7FH)
સ્કેચ પેડ	30H-7FH	જનરલ પર્પઝ
SFRs	80H-FFH	કંટ્રોલ રજિસ્ટર્સ

સાચાચામ:

8051 Internal RAM (128 bytes):

```
+-----+ 00H
| Register Bank 0 |
+-----+ 08H
| Register Bank 1 |
+-----+ 10H
| Register Bank 2 |
+-----+ 18H
| Register Bank 3 |
+-----+ 20H
| Bit-addressable |
+-----+ 30H
|                  |
| Scratch Pad      |
|                  |
+-----+ 80H
```

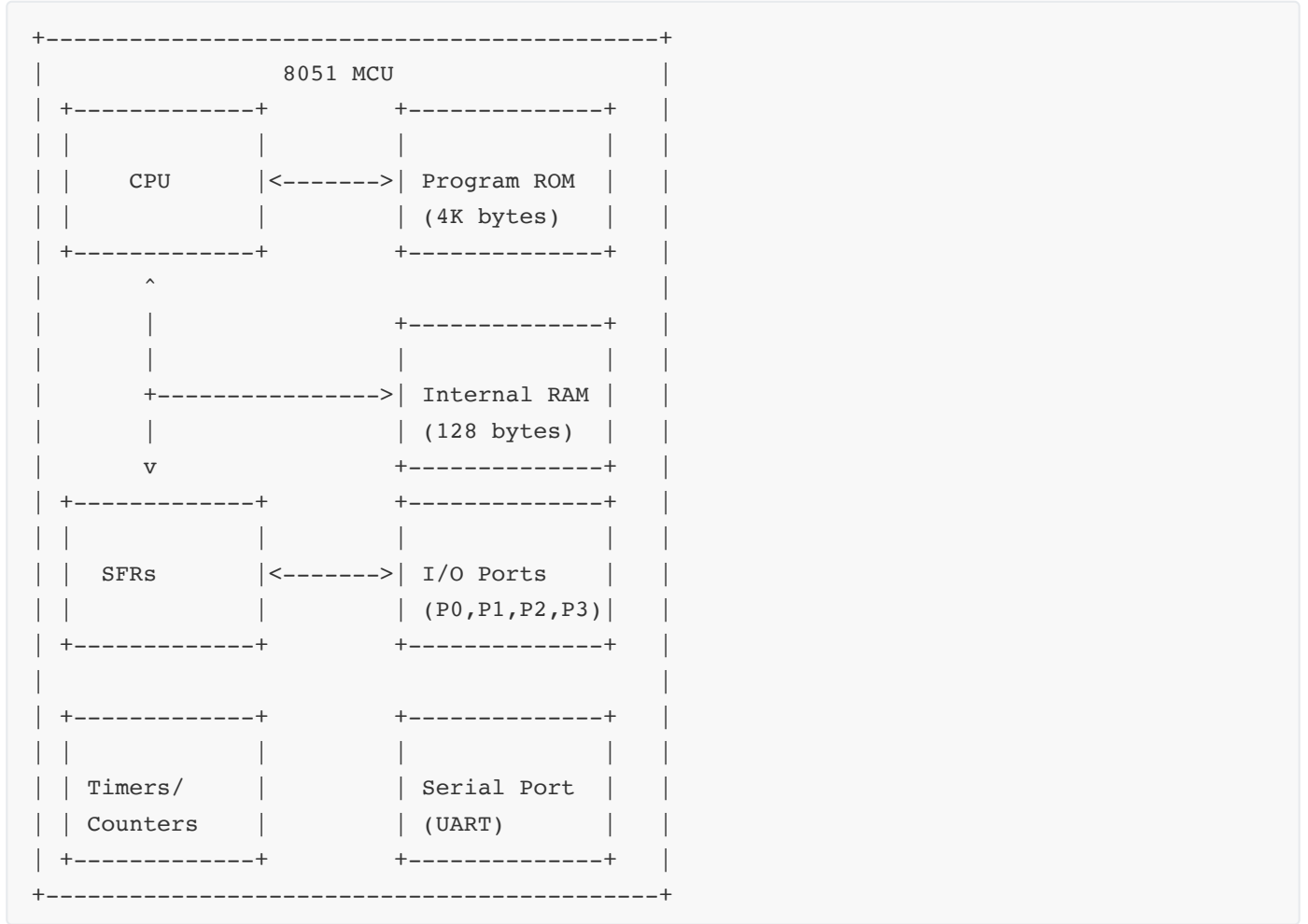
યાદ રાખવા માટે: "RBBS: રજિસ્ટર્સ-બિટ્સ-બફર-સ્કેચ"

પ્રશ્ન 3(ક OR) [7 ગુણ]

આકૃતિની મદદથી 8051 માઇક્રોકંટ્રોલરનું આર્કિટેક્ચર સમજાવો.

જવાબ:

કમ્પોનન્ટ	કાર્ય
CPU	8-બિટ પ્રોસેસર ALU સાથે
મેમરી	4K ROM, 128 બાઇટ્સ RAM
I/O પોર્ટ્સ	ચાર 8-બિટ પોર્ટ્સ (P0-P3)
ટાઇમર્સ	બે 16-બિટ ટાઇમર/કાઉન્ટર
સીરિયલ પોર્ટ	ફુલ-ડુપ્લેક્સ UART
ઇન્ટરપ્ટ	પાંચ ઇન્ટરપ્ટ સોર્સ
સ્પેશિયલ ફંક્શન રજિસ્ટર્સ	કંટ્રોલ રજિસ્ટર્સ

સાચાગ્રામ:

- **હાર્ડવેર્ડ આર્કિટેક્ચર:** અલગ પ્રોગ્રામ અને ડેટા મેમરી
- **CISC ડિઝાઇન:** સમૃદ્ધ ઇન્સ્ટ્રક્શન સેટ (100થી વધુ સૂચનાઓ)
- **બિલ્ટ-ઇન પેરિફેરલ્સ:** બાહ્ય કમ્પોનન્ટ્સની જરૂર નથી
- **સિંગલ-ચિપ સોલ્યુશન:** એક જ ચિપ પર સંપૂર્ણ સિસ્ટમ

યાદ રાખવા માટે: "CAPITALS: CPU આર્કિટેક્ચર પોર્ટ્સ I/O ટાઇમર ALU ઇન્ટરફેસ સીરિયલ"

પ્રશ્ન 4(અ) [3 ગુણ]

બાહ્ય RAM સ્થાન 0123h થી TL0 અને બાહ્ય RAM સ્થાન 0234h થી TH0 ડેટાને કોપી કરવા માટે 8051 એસેમ્બલી લેંગ્વેજ પ્રોગ્રામ લખો.

જવાબ:

```
MOV DPTR, #0123H ; DPTR મા સોર્સ એડ્રેસ 0123H લોડ કરો
MOVB A, @DPTR    ; બાહ્ય RAM માંથી ડેટા વાંચો
MOV TL0, A        ; ટાઇમર 0 લે બાઇટમાં કોપી કરો

MOV DPTR, #0234H ; DPTR મા સોર્સ એડ્રેસ 0234H લોડ કરો
MOVB A, @DPTR    ; બાહ્ય RAM માંથી ડેટા વાંચો
MOV TH0, A        ; ટાઇમર 0 હાઈ બાઇટમાં કોપી કરો
```

મુખ્ય સ્ટેપ્સ:

- બાહ્ય RAM એક્સેસ માટે DPTR વાપરો
- બાહ્ય મેમરી એક્સેસ માટે MOVX સૂચના
- ટાઇમર રજિસ્ટર્સમાં સીધો ટ્રાન્સફર

યાદ રાખવા માટે: "DRAM: DPTR વાંચો એક્સેસ હલાવો"

પ્રશ્ન 4(બ) [4 ગુણ]

પોર્ટ P1.3 પર ઇન્ટરફેસ કરેલ LED ને 1ms ના સમય અંતરાલ પર બ્લિંક કરવા માટે 8051 એસેમ્બલી લેંગ્વેજ પ્રોગ્રામ લખો.

જવાબ:

```

AGAIN:  SETB P1.3      ; P1.3 પર LED ચાલુ કરો
        ACALL DELAY    ; ડિલે સબરૂટીન કોલ કરો
        CLR  P1.3      ; P1.3 પર LED બંધ કરો
        ACALL DELAY    ; ડિલે સબરૂટીન કોલ કરો
        SJMP AGAIN     ; હંમેશા રિપીટ કરો

DELAY:   MOV  R7, #250  ; આઉટર લૂપ માટે R7 લોડ કરો
OUTER:   MOV  R6, #1    ; ઇનર લૂપ માટે R6 લોડ કરો
INNER:   DJNZ R6, INNER ; R6 ઝીરો થાય ત્યાં સુધી ઘટાડો
        DJNZ R7, OUTER ; R7 ઝીરો થાય ત્યાં સુધી ઘટાડો
        RET            ; સબરૂટીનમાંથી પાછા ફરો
  
```

મુખ્ય સ્ટેપ્સ:

- LED બ્લિંક કરવા માટે P1.3 પીન ટોગલ કરો
- ટાઇમિંગ માટે નેસ્ટેડ ડિલે લૂપ
- સતત બ્લિંકિંગ માટે અનંત લૂપ

યાદ રાખવા માટે: "STACI: સેટ-ટાઇમર-એન્ડ-ક્લિયર-ઇન્ફિનિટલી"

પ્રશ્ન 4(ક) [7 ગુણ]

8051 માઇક્રોકંટ્રોલરના એડ્રેસિંગ મોડ્સની યાદી બનાવો અને ઉદાહરણની મદદથી તે બધાને સમજાવો.

જવાબ:

એડ્રેસિંગ મોડ	ઉદાહરણ	વર્ણન
ઇમીડિયેટ	MOV A, #25H	ડેટા સૂચનામાં છે
રજિસ્ટર	MOV A, R0	ડેટા રજિસ્ટરમાં છે
ડાયરેક્ટ	MOV A, 30H	ડેટા RAM એડ્રેસ પર છે
ઇન્ડાયરેક્ટ	MOV A, @R0	R0/R1 એડ્રેસ ધરાવે છે
ઇન્ડેક્સ	MOVC A, @A+DPTR	પ્રોગ્રામ મેમરી એક્સેસ
બિટ	SETB P1.3	વ્યક્તિગત બિટ્સ એક્સેસ
રિલેટિવ	SJMP LABEL	8-બિટ ઑફસેટ સાથે જમ્પ

ઉદાહરણો:

- **ઇમીડિયેટ:** MOV A, #55H (A માં 55H લોડ કરો)
- **રજિસ્ટર:** ADD A, R3 (A માં R3 ઉમેરો)
- **ડાયરેક્ટ:** MOV 40H, A (A ને એડ્રેસ 40H પર સ્ટોર કરો)
- **ઇન્ડાયરેક્ટ:** MOV @R0, #5 (R0 માં રહેલા એડ્રેસ પર 5 સ્ટોર કરો)
- **ઇન્ડેક્સ:** MOVC A, @A+DPTR (ડોસ મેમરી વાંચો)
- **બિટ:** CLR C (કેરી ફ્લેગ સાફ કરો)
- **રિલેટિવ:** JZ LOOP (જો A ઝીરો હોય તો જમ્પ કરો)

યાદ રાખવા માટે: "I'M DIRBI: ઇમીડિયેટ રજિસ્ટર ડાયરેક્ટ બિટ ઇન્ડેક્સ"

પ્રશ્ન 4(અ OR) [3 ગુણ]

RAM સ્થાન 14h માંથી RAM સ્થાન 11h નાં ડેટાને બાદ કરવા માટે 8051 એસેમ્બલી લેંગ્વેજ પ્રોગ્રામ લખો; RAM સ્થાન 3Ch માં પરિણામ મૂકો.

જવાબ:

```
MOV A, 14H ; RAM લોકેશન 14H નો કન્ટેન્ટ A માં લોડ કરો
CLR C ; કેરી ફ્લેગ સાફ કરો
SUBB A, 11H ; બોરો સાથે 11H ના કન્ટેન્ટ બાદ કરો
MOV 3CH, A ; પરિણામને RAM લોકેશન 3CH માં સ્ટોર કરો
```

મુખ્ય સ્ટેપ્સ:

- એક્યુમ્યુલેટરમાં મિન્યુએન્ડ લોડ કરો
- સાચા સબટ્રેક્શન માટે કેરી સાફ કરો
- બોરો સાથે સબટ્રેક્શન માટે SUBB વાપરો
- પરિણામને ડેસ્ટિનેશનમાં સ્ટોર કરો

યાદ રાખવા માટે: "LCSS: લોડ-ક્લિયર-સબટ્રેક્ટ-સ્ટોર"

પ્રશ્ન 4(બ OR) [4 ગુણ]

મોડ 1 માં ટાઇમર 0 નો ઉપયોગ કરીને પોર્ટ 1 ના બીટ 3 પર 50% ડ્યુટી સાયકલની સ્ક્વેર વેવ જનરેટ કરવા માટે 8051 એસેમ્બલી લેંગ્વેજ પ્રોગ્રામ લખો.

જવાબ:

```
MOV TMOD, #01H ; ટાઇમર 0, મોડ 1 (16-બિટ)
AGAIN: MOV TH0, #0FCH ; હાઈ બાઇટ લોડ કરો
MOV TL0, #18H ; લો બાઇટ લોડ કરો (-1000 16-બિટમાં)
SETB TR0 ; ટાઇમર ચાલુ કરો
JNB TF0, $ ; ઓવરફ્લો માટે રાહ જુઓ
CLR TR0 ; ટાઇમર બંધ કરો
CLR TF0 ; ટાઇમર ફ્લેગ સાફ કરો
CPL P1.3 ; P1.3 ટોગલ કરો
SJMP AGAIN ; રિપીટ કરો
```

મુખ્ય સ્ટેપ્સ:

- મોડ 1 માં ટાઇમર 0 કોન્ફિગર કરો
- 1ms ડિલે માટે ટાઇમરમાં વેલ્યુ પ્રીલોડ કરો
- ટાઇમર ઓવરફ્લો માટે રાહ જુઓ
- સ્ક્વેર વેવ માટે આઉટપુટ બિટ ટોગલ કરો

યાદ રાખવા માટે: "MSTCCS: મોડ-સેટ-ટાઇમર-ચેક-ક્લિયર-સ્વિચ"

પ્રશ્ન 4(ક OR) [7 ગુણ]

8051 માઇક્રોકન્ટ્રોલર માટે કોઈપણ સાત લોજિકલ ઈન્સ્ટ્રક્શન ઉદાહરણ સાથે સમજાવો.

જવાબ:

ઈન્સ્ટ્રક્શન	ઉદાહરણ	ઓપરેશન
ANL	ANL A, #3FH	લોજિકલ AND
ORL	ORL P1, #80H	લોજિકલ OR
XRL	XRL A, R0	લોજિકલ XOR
CLR	CLR A	ક્લિયર (0 સેટ)
CPL	CPL P1.0	કોમ્પ્લિમેન્ટ (ઇન્વર્ટ)
RL	RL A	રોટેટ લેફ્ટ
RR	RR A	રોટેટ રાઇટ

ઉદાહરણો:

- ANL:** ANL A, #0FH (A = A AND 0FH, હાઈ નિબલ માસ્ક)

- **ORL:** ORL 20H, A (20H = 20H OR A, બિટ્સ સેટ)
- **XRL:** XRL A, #55H (A = A XOR 55H, બિટ્સ ટોગલ)
- **CLR:** CLR C (કેરી ફ્લેગ ક્લિયર, C = 0)
- **CPL:** CPL A (A ને કોમ્પ્લિમેન્ટ, A = NOT A)
- **RL:** RL A (A ને એક બિટ લેફ્ટ રોટેટ)
- **RR:** RR A (A ને એક બિટ રાઇટ રોટેટ)

યાદ રાખવા માટે: "A-OX-CCR: AND OR XOR ક્લિયર કોમ્પ્લિમેન્ટ રોટેટ"

પ્રશ્ન 5(અ) [3 ગુણ]

8051 માઇક્રોકંટ્રોલર સાથે Push button Switch નું ઇન્ટરફેસિંગ દોરો.

જવાબ:



કમ્પોનન્ટ	કનેક્શન
પુશ બટન	P1.0 અને GND વચ્ચે
પુલ-અપ રેસિસ્ટર	P1.0 અને VCC વચ્ચે 10K
પોર્ટ પિન	P1.0 ઇનપુટ તરીકે કોન્ફિગર

મુખ્ય પોઇન્ટ્સ:

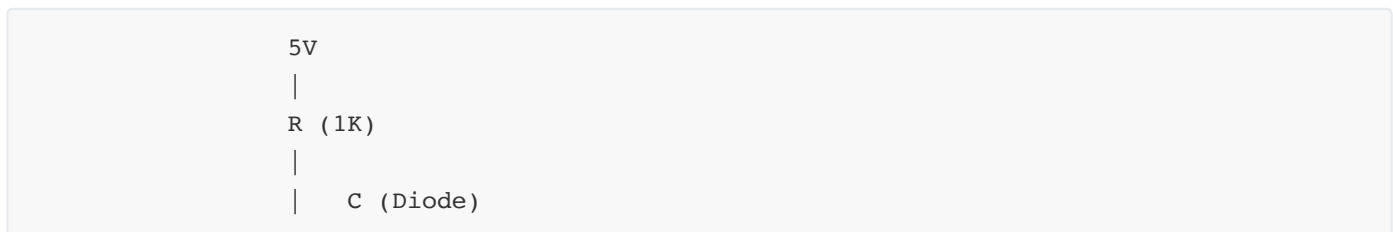
- એક્ટિવ-લો કોન્ફિગરેશન (બટન દબાવવાથી 0 મળે)
- પુલ-અપ રેસિસ્ટર ફ્લોટિંગ ઇનપુટ રોકે
- કોઈપણ I/O પિન સાથે જોડી શકાય

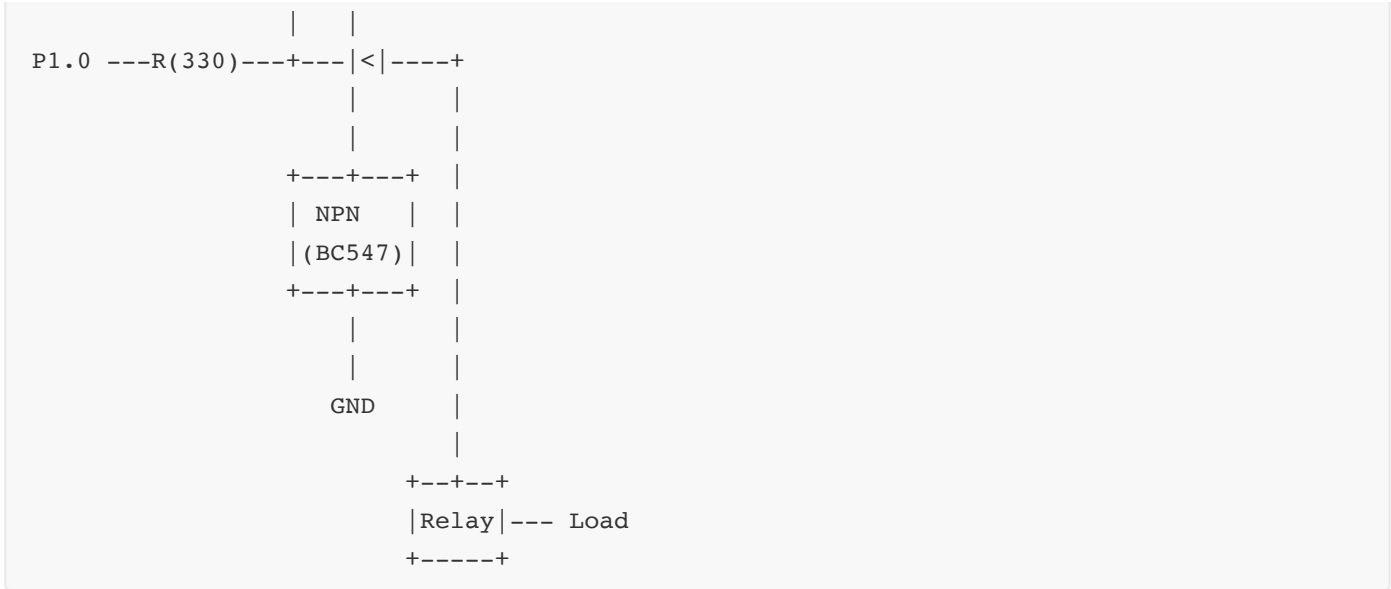
યાદ રાખવા માટે: "PIP: પુલ-અપ-ઇનપુટ-પ્રેસ"

પ્રશ્ન 5(બ) [4 ગુણ]

8051 માઇક્રોકંટ્રોલર સાથે રિલે ઇન્ટરફેસ કરો.

જવાબ:





કમ્પોનન્ટ	હેતુ
NPN ટ્રાન્ઝિસ્ટર	કરંટ એમ્પ્લિફિકેશન
ડાયોડ	બેક EMF પ્રોટેક્શન
રેસિસ્ટર્સ	કરંટ લિમિટિંગ
રિલે	હાઈ-પાવર સ્વિચિંગ

મુખ્ય સ્ટેપ્સ:

- પોર્ટ પિન ટ્રાન્ઝિસ્ટર બેઝ ડ્રાઇવ કરે
- ટ્રાન્ઝિસ્ટર રિલે કોઇલ સ્વિચ કરે
- ડાયોડ બેક EMF સામે રક્ષણ આપે
- રિલે કોન્ટેક્ટ હાઈ-પાવર લોડ સ્વિચ કરે

યાદ રાખવા માટે: "TRIP: ટ્રાન્ઝિસ્ટર-રિલે-ઇન્ટરફેસ-પ્રોટેક્શન"

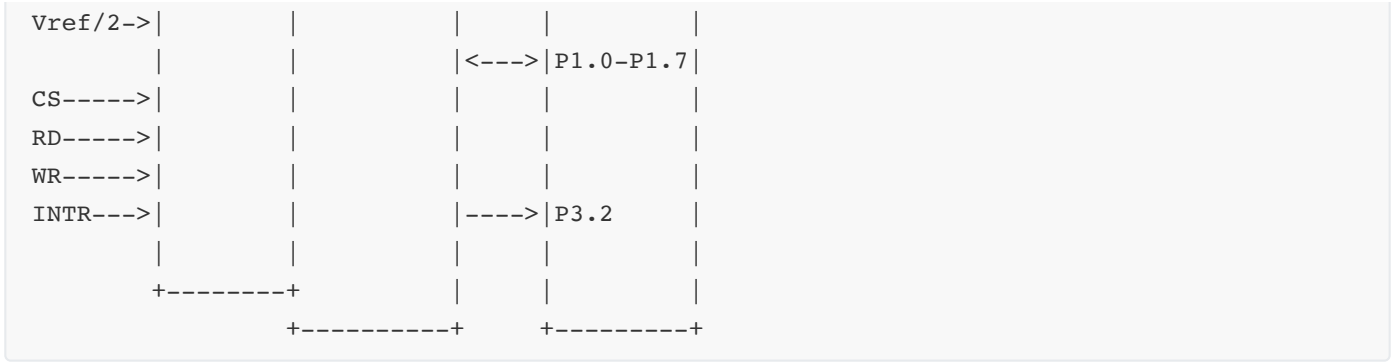
પ્રશ્ન 5(ક) [7 ગુણ] (ચાલુ)

8051 માઇક્રોકંટ્રોલર સાથે ADC0804 ઇન્ટરફેસ કરો.

જવાબ:

સર્કિટ ડાયાગ્રામ:





કનેક્શન	8051 પિન	ADC0804 પિન
ડેટા બસ	P1.0-P1.7	D0-D7
CS	P3.0	CS
RD	P3.1	RD
WR	P3.2	WR
INTR	P3.3	INTR

- **ADC0804:** 8-બિટ A/D કન્વર્ટર 0-5V ઇનપુટ રેન્જ સાથે
- **ઇન્ટરફેસ:** ડેટા પિન પોર્ટ 1 સાથે, કંટ્રોલ પોર્ટ 3 સાથે જોડો
- **ઓપરેશન:** કન્વર્ઝન શરૂ કરવા ADC ને લખો, INTR માટે રાહ જુઓ, રિઝલ્ટ વાંચો
- **રેઝોલ્યુશન:** 8-બિટ (256 સ્ટેપ) 0-5V માટે ~19.5mV પ્રતિ સ્ટેપ

યાદ રાખવા માટે: "CRIW: કંટ્રોલ-રીડ-ઇન્ટરફેસ-રાઇટ"

પ્રશ્ન 5(અ OR) [3 ગુણ]

વિવિધ ક્ષેત્રોમાં માઇક્રોકંટ્રોલરની એપ્લિકેશનોની સૂચિ બનાવો.

જવાબ:

ક્ષેત્ર	એપ્લિકેશન્સ
ઔદ્યોગિક	મોટર કંટ્રોલ, ઓટોમેશન, PLCs
મેડિકલ	પેશન્ટ મોનિટરિંગ, ડાયગ્નોસ્ટિક ઉપકરણો
કન્ઝ્યુમર	વોશિંગ મશીન, માઇક્રોવેવ, રમકડાં
ઓટોમોટિવ	એન્જિન કંટ્રોલ, ABS, એરબેગ સિસ્ટમ
કમ્યુનિકેશન	મોબાઇલ ફોન, મોડેમ, રાઉટર
સિક્યુરિટી	એક્સેસ કંટ્રોલ, અલાર્મ સિસ્ટમ

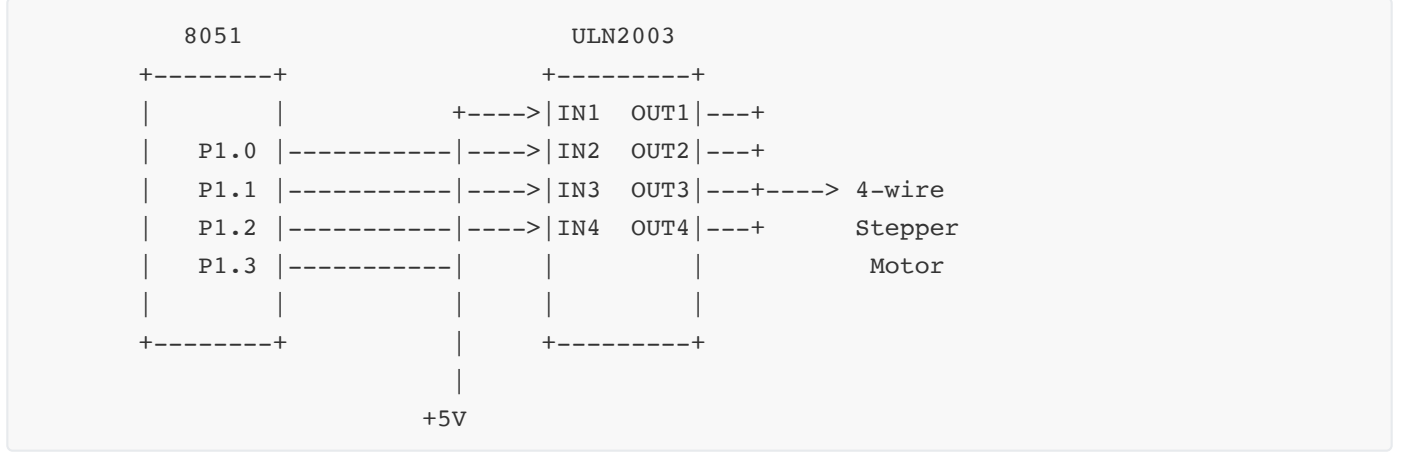
યાદ રાખવા માટે: "I-MACS: ઇન્ડસ્ટ્રિયલ-મેડિકલ-ઓટોમોટિવ-કન્ઝ્યુમર-સિક્યુરિટી"

પ્રશ્ન 5(બ OR) [4 ગુણ]

8051 માઇક્રોકંટ્રોલર સાથે સ્ટેપર મોટર ઇન્ટરફેસ કરો.

જવાબ:

સર્કિટ ડાયાગ્રામ:



કમ્પોનન્ટ	હેતુ
ULN2003	ડ્રાઇવર IC ડાર્લિંગટન એરે સાથે
પોર્ટ પિન	P1.0-P1.3 4 મોટર ફેઝ માટે
પાવર સપ્લાય	મોટર માટે અલગ સપ્લાય

કોડ સ્ટ્રક્ચર:

```

; ક્લોકવાઇઝ રોટેશન સિક્વન્સ
STEP_SEQ: DB 0000_1000B ; સ્ટેપ 1
           DB 0000_1100B ; સ્ટેપ 2
           DB 0000_0100B ; સ્ટેપ 3
           DB 0000_0110B ; સ્ટેપ 4

```

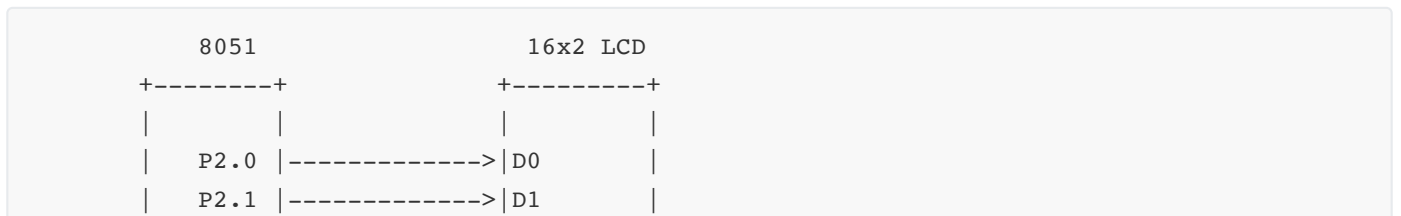
યાદ રાખવા માટે: "PDCS: પોર્ટ-ડ્રાઇવર-કન્ટ્રોલ-સિક્વન્સ"

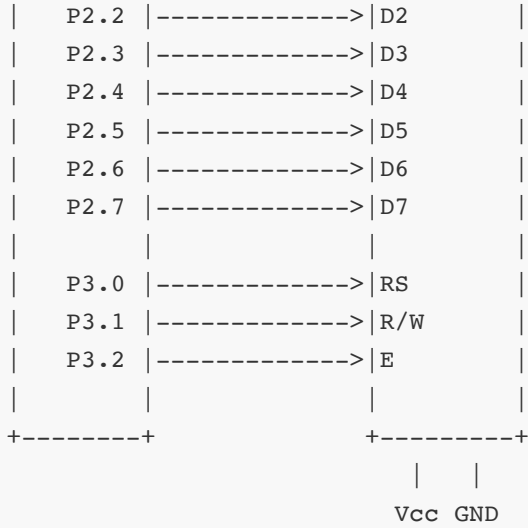
પ્રશ્ન 5(ક OR) [7 ગુણ]

8051 માઇક્રોકંટ્રોલર સાથે LCD ઇન્ટરફેસ કરો.

જવાબ:

સર્કિટ ડાયાગ્રામ:





કનેક્શન	હેતુ
ડેટા પિન (D0-D7)	P2.0-P2.7 સાથે જોડો
RS	રજિસ્ટર સિલેક્ટ (0=કમાન્ડ, 1=ડેટા)
R/W	રીડ/રાઇટ (0=રાઇટ, 1=રીડ)
E	એનેબલ સિગ્નલ (એક્ટિવ હાઇ)

બેઝિક કમાન્ડ્સ:

0x01 - ડિસ્પ્લે ક્લિયર
 0x02 - હોમ પોઝિશન
 0x0C - ડિસ્પ્લે ON, કર્સર OFF
 0x38 - 8-બિટ, 2 લાઇન, 5x7 ડોટ્સ

- **ઇનિશિયલાઇઝેશન:** LCD ને 8-બિટ મોડ, 2 લાઇન માટે કોન્ફિગર કરો
- **રાઇટિંગ:** RS=1 સાથે ડેટા, RS=0 સાથે કંટ્રોલ મોકલો
- **ટાઇમિંગ:** E પલ્સ ટાઇમિંગ જરૂરિયાતો પૂરી કરવી જોઈએ
- **કોન્ટ્રાસ્ટ:** VEE પિન પર પોટેન્શિયોમીટર સાથે એડજસ્ટ કરો

યાદ રાખવા માટે: "DICE: ડેટા-ઇન્ટ્રક્શન-કંટ્રોલ-એનેબલ"