## ৫.৭ ইন্টারাপ্টের সময় রেজিস্টার সুরক্ষাকরণ বর্ণনা

**(Describe the Register Protection During Interrupt)**

এটি Interrupt handler বা রুটিনে একটি অত্যন্ত গুরুত্বপূর্ণ নিয়ম হলো ইন্টারাপ্ট সৃষ্টির সময় যে অবস্থায় ছিল তাকে পুনরায় সেভাবে ফেরত আনা। অর্থাৎ যদি কোন ইন্টারাপ্ট রেজিস্টার (যথা, এক্যুমুলেটর) ব্যবহার করে, তবে এটি অবশ্যই নিশ্চিত করতে হবে যে, রেজিস্টারের মান ইন্টারাপ্টের শুরুতে যা ছিল শেষেও তাই থাকে। ইন্টারাপ্ট রুটিন সম্পাদনকালে অনেক সময় (বিশেষ করে MOV নির্দেশনায় রেজিস্টার মান পরিবর্তন হয়ে যায়। সাধারণত ইন্টারাপ্টের সময় স্টাক (stack) এ PUSH ও POP অনুক্রম দ্বারা রেজিস্টারের মানকে সুরক্ষা করা হয়।

ইন্টারাপ্ট রুটিনকে নিম্নবর্ণিত রেজিস্টারসমূহকে অবশ্যই সুরক্ষা করতে হবে-

1. PSW
2. PSW
3. B
4. DPTR (DPH/DPL)
5. ACC
6. Registers R0-R7

এখানে, PSW রেজিস্টারে অনেক স্বতন্ত্র বিট আছে, যাদেরকে 8051 এর বিভিন্ন নির্দেশনা দ্বারা সেট করা যায়। সেক্ষেত্রে ইন্টারাপ্টের শুরুতে এবং শেষে Stack এ Pushing ও Popping করে PSW রেজিস্টারকে সর্বদা সুরক্ষা করা একটি উত্তম পরিকল্পনা।

PUSH R0 কোডটিকে অধিকাংশ অ্যাসেম্বলার বিশ্লেষণ করে না।

কারণ R0 কোন প্রকারভেদে চিকানা যায়, যাতে PUSH ও POP নির্দেশনা ব্যবহার করা যায়। R0 অভ্যন্তরীণ মেমোরি চিকানা 00h, 08h, 10h বা 18 কে অর্পণ করতে পারে (refer)। তাই ইন্টারাপ্ট রুটিনে কোন "R" রেজিস্টার ব্যবহার হলে এটে স্টাকে পরম চিকানা (absolute address) কে PUSH করতে হয়। তাই PUSH R0 এর পরিবর্তে PUSH 00h সম্পাদন করা হয়। তবে এটি default রেজিস্টারের ক্ষেত্রে প্রযোজ্য। যদি বিকল্প রেজিস্টার নির্বাচন করা হয়, তবে ঐ রেজিস্টারের সঙ্কিপূর্ণ চিকানায় PUSH করতে হবে।

## ৫.৮ বাহ্যিক হার্ডওয়্যার, টাইমার ও সিরিয়াল কমিউনিকেশন ইন্টারাপ্ট বর্ণনা

**(Describe External Hardware, Timer and Serial Communication Interrupt)**

**বাহ্যিক হার্ডওয়্যার ইন্টারাপ্ট (External hardware interrupt) :**

8051 এর INT0 (পিন 12) বা INT1 (পিন 13) পিনের একটি Low level বা Negative edge এর ফলে বাহ্যিক ইন্টারাপ্ট ঘটে। এই ইন্টারাপ্টের জন্য উৎপন্ন flag bit গুলো হলো : যথাক্রমে IE0 ও IE1 (TCON রেজিস্টার)। এবং যদি ইন্টারাপ্ট Transition activated (Negative edge) হয়, এদেরকে হার্ডওয়্যার দ্বারা clear করা হয়। যদি ইন্টারাপ্ট Level-activated হয়, তবে মাইক্রোকন্ট্রোলারের অভ্যন্তরীণ হার্ডওয়্যারের পরিবর্তে বাহ্যিক ইন্টারাপ্ট উৎস Request flag এর স্বেচ্ছাকে নিয়ন্ত্রণ করে।

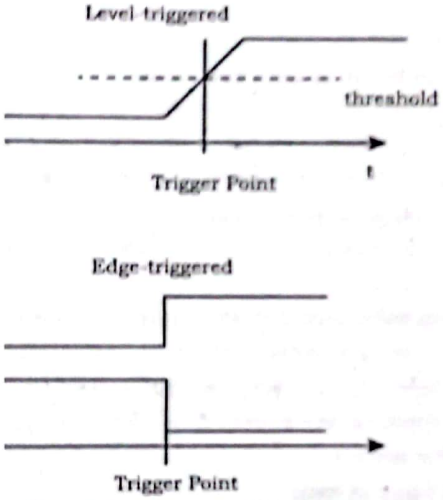
TCON রেজিস্টারের IT0 ও IT1 এর মাধ্যমে Low level activated ইন্টারাপ্ট বা negative-edge activated ইন্টারাপ্ট নির্বাচন প্রোগ্রাম করা হয়। যদি IT1 = 0 হয়, তবে INT1 পিনে একটি Low level দ্বারা External interrupt 1 এ সূচনাপ্ত হয়। যদি IT1 = 1 হয়, তবে External interrupt 1 edge-triggered হয়। এই মুডে যদি INT1 পিনে এক সাইকেলে High এবং পরবর্তী সাইকেলে Low দেখায় তবে TCON রেজিস্টারের IE1 flag বিটটি সেট হয় এবং এর Interrupt request করে। যেহেতু বাহ্যিক ইন্টারাপ্ট এক মেশিন সাইকেলে একবার স্যাম্পল হয়, তাই সঠিক স্যাম্পলিং নিশ্চিতকরণের জন্য ন্যূনতম 12 ক্লক পিরিয়ডে একবার ইনপুট হওয়া উচিত। যদি বাহ্যিক ইন্টারাপ্ট Transition activated হয়, তবে বাহ্যিক উৎস request pin কে অন্তত এক সাইকেলের জন্য অবশ্যই high রাখবে এবং পরে Transition শনাক্তকরণ নিশ্চিতকরণের জন্য আরও এক সাইকেলের জন্য request পিনকে Low রাখবে। যখন CPU ইন্টারাপ্টকে সেবা করে তখন IE0 ও IE1 স্বয়ংক্রিয়ভাবে Clear হয়।

যদি বাহ্যিক ইন্টারাপ্ট level-activated হয়, তবে বাহ্যিক উৎস অবশ্যই request কে কার্যকরী রাখবে যতক্ষণ পর্যন্ত না requested interrupt প্রকৃতপক্ষে উৎপন্ন হয়। পরে ISR শেষ হওয়ার পূর্বে একে অবশ্যই নিষ্ক্রিয় করতে হবে নতুবা অন্য ইন্টারাপ্ট উৎপন্ন হবে। পরবর্তীর ISR এর একটি ক্রিয়া ইন্টারাপ্ট উৎসকে ইন্টারাপ্টিং লিপ্তালের নিষ্ক্রিয় অবস্থায় নিয়ে যায়।



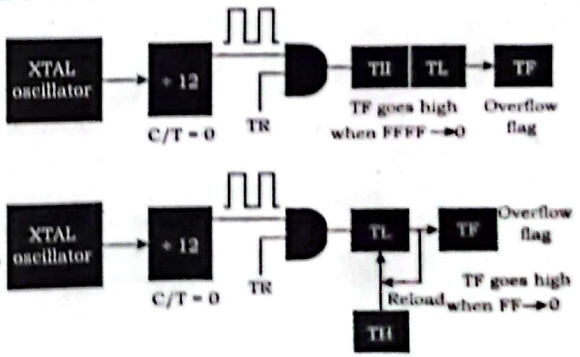
**টাইমার ইন্টারাপ্ট (Timer Interrupt) :** 8051 মাইক্রোকন্ট্রোলারের T0 ও T1 নামের 16 বিটের দুটি টাইমার/কাউন্টার রয়েছে। এরা টাইম ডিলে উৎপন্ন করার জন্য বা মাইক্রোকন্ট্রোলারে বহিরস্থ ঘটনা (event) এর কাউন্টার বা event counter হিসেবে ব্যবহার করা হয়। যেহেতু 8051 এর আর্কিটেকচার 8 বিটের, তাই 16 বিটের টাইমার access করার জন্য প্রত্যেকের জন্য দুটি করে 8 বিটের রেজিস্টার রয়েছে। এরা হলো TL0, TH0, TL1 ও TH1। প্রথম 0 ও 1 যথাক্রমে Timer 0 ও Timer 1 প্রকাশ করে।

উভয় টাইমার বিভিন্ন টাইমার অপারেশন মুড সেটিং এর জন্য একই রেজিস্টার TMOD ব্যবহার করে। TMOD একটি 8 বিটের রেজিস্টার, যার নিচের 4 বিট Timer 0 এবং উপরের 4 বিট Timer1 এর জন্য বরাদ্দকৃত। টাইমার অপারেশনকে নিয়ন্ত্রণের জন্য অপর একটি বিশেষ রেজিস্টার TCON ব্যবহার করা হয়।




চিত্র-৫.২ : Level-trigger ও Edge-trigger

8051 টাইমার সর্বদা count up করে। যখন তাদের কাউন্ট সর্বোচ্চ কাউন্ট থেকে 0000 আবর্তন করে, তখন TCON রেজিস্টারে তাদের টাইমার flag (TF0 বা TF1) উৎপন্ন করে। ব্যবহারকারীর প্রোগ্রাম দ্বারা TCON রেজিস্টারের টাইমার Run flag বিট TR0 বা TR1 সেট করে টাইমার অপারেশন শুরু করা হয়। TF0 বা TF1 সেট হলেই টাইমার ইন্টারাপ্ট ঘটে।




চিত্র-৫.৩ : টাইমার অপারেশন

**সিরিয়াল ইন্টারাপ্ট (Serial interrupt) :**

যখন সিরিয়াল পোর্ট কর্তৃক একটি বাইট গৃহীত হয়, তখন RI ইন্টারাপ্ট flag এবং পোর্ট থেকে এক বাইট প্রেরিত হলে তখন TI ইন্টারাপ্ট flag সেট হয়। RI ও TI এর যে কোনটি সেট হলে একটি সিরিয়াল ইন্টারাপ্ট Trigger হয়। Transmit interrupt (TI) হয় যখন SBUF রেজিস্টারে প্রেরিতব্য অক্ষরের লিখন শেষ হয় এবং Receive interrupt হয় যখন কোন অক্ষর সম্পূর্ণভাবে গৃহীত হয় এবং SBUF রেজিস্টারে পঠনের জন্য অপেক্ষা করে। সিরিয়াল পোর্ট ইন্টারাপ্ট, সিরিয়াল ইন্টারাপ্ট অন্যান্য ইন্টারাপ্টের চেয়ে সামান্য ভিন্ন, এর Flag হার্ডওয়্যার দ্বারা clear হয় না। কারণ একটি সিরিয়াল পোর্ট ইন্টারাপ্টের জন্য দুটি উৎস RI ও TI। ইন্টারাপ্টের উৎস অবশ্যই ISR এ নির্ধারণ করতে হয় এবং Interrupting flag সফটওয়্যার দ্বারা clear করা হয়।

## ৫.৯ ইন্টারাপ্টের সাধারণ সমস্যা বর্ণনা

## (State the Common Problem With Interrupt)

ইন্টারাপ্ট একটি শক্তিশালী হাতিয়ার হিসেবে 8051 প্রোগ্রাম উন্নয়ন করে কিন্তু এর অশুদ্ধ ব্যবহার প্রোগ্রামে ব্যাপক ত্রুটি আসে, যা নির্ণয় ও সংশোধন খুবই কষ্টসাধ্য। যদি প্রোগ্রামে ইন্টারাপ্ট থাকে এবং প্রোগ্রাম ক্রাশ করে বা আকাঙ্ক্ষিত কাজ সম্পাদন না করে, তবে নিম্নবর্ণিত ইন্টারাপ্ট-সম্পর্কিত বিষয়গুলো পর্যালোচনা করা হয় :

**(১) রেজিস্টার সুরক্ষাকরণ (Register protection)**

প্রোগ্রামে রেজিস্টার সুরক্ষা নিশ্চিত হতে হবে। যদি প্রোগ্রামে রেজিস্টার সুরক্ষা করা না হয়, তবে অনাকাঙ্ক্ষিত ফলাফল হতে পারে।

**(২) সুরক্ষিত মান পূর্বাবস্থায় ফিরিয়ে আনতে ভুলে যাওয়া (Forgetting to restore protected value) :**

অপর এক সাধারণ সমস্যা হলো রেজিস্টারসমূহকে Stack এ Push করা এবং ইন্টারাপ্ট থেকে বের হওয়ার পূর্বে তা Stack থেকে POP করতে ভুলে যাওয়া। ফলে একটি অতিরিক্ত মান Stack এ থেকে যায়। যখন RETI নির্দেশনা সম্পাদন হয়, তখন প্রত্যাবর্তন ঠিকানা সঠিক হলেও 8051 এই মানকে ব্যবহার করে। এতে প্রোগ্রাম প্রায়ই নিশ্চিত ক্রাশ করে।

**(৩) RETI এর পরিবর্তে RET এর ব্যবহার :**

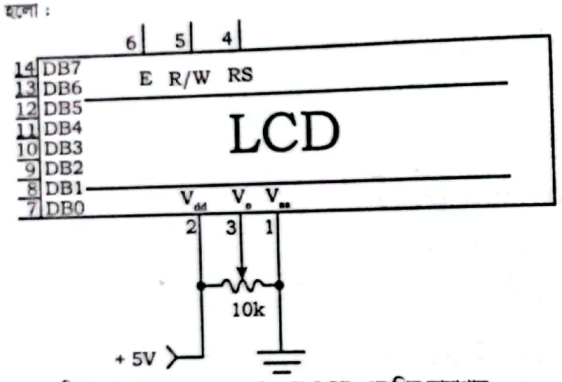
ইন্টারাপ্ট সবসময় RETI নির্দেশনা দ্বারা শেষ হয়। RET নির্দেশনা ইন্টারাপ্টের শেষ করে না। RETI এর পরিবর্তে RET ব্যবহার করলে মূল প্রোগ্রাম স্বাভাবিকভাবে চলছে মনে হয়, কিন্তু ইন্টারাপ্ট শুধুমাত্র একবার সম্পাদিত হতে পারে এবং ইন্টারাপ্ট সম্পাদন হঠাৎ করে থেমে যেতে পারে।

CamScanner

# অধ্যায় ৬ এলসিডি ও কী-বোর্ড ইন্টারফেসিং
# Under Stand LCD and Keyboard Interfacing

## ৬.১ এলসিডি-এর পিন ডায়াগ্রাম বর্ণনা
**(Describe the Pin Diagram of LCD)**

আধুনিক যুগে LCD মনিটর, টেলিভিশন ব্যাপক সমাদৃত। LCD (Liquid Crystal Display) পর্দা একটি বহুল ব্যবহৃত ইলেকট্রনিক ডিসপ্লে মডিউল। 16 × 2 LCD ডিসপ্লে একটি অত্যন্ত মৌলিক মডিউল, যা বিভিন্ন ডিভাইস ও সার্কিটে সাধারণত অধিক ব্যবহার করা হয়। 16 × 2 LCD বলতে এটি প্রতি লাইনে 16টি করে অক্ষর নিয়ে 2 লাইন প্রদর্শন করতে পারে। এর প্রতিটি অক্ষর 5 × 6 পিক্সেল মেট্রিক্স প্রদর্শন করে। নিচে একটি 16 × 2 LCD, HD44780-এর পিন ডায়াগ্রাম দেখানো হলো :




চিত্র-৬.১ : HD44780 (16 × 2) LCD-এর পিন ডায়াগ্রাম

**একটি HD44780 (16 × 2) LCD-এর পিনের বর্ণনা :**

i. DB0 – DB7 : 4 বিট ডাটা বাস, ডাটা লিখন/পঠন করে।
ii. Vss, Vdd : ভোল্টেজ সাপ্লাই পিন।
iii. R/W : LCD তে লিখন বা LCD হতে পঠন সিগন্যাল পিন।
iv. RS : রেজিস্টার নির্বাচন পিন। এর সাহায্যে ইনস্ট্রাকশন রেজিস্টার ও ডাটা রেজিস্টার নির্বাচন করা হয়।
v. E : এনাবল পিন। এটি LCD তে লিখন/পঠন (R/W) অপারেশন কার্যকর বা অকার্যকর করে। এটি High হলে কার্যকর হয় এবং Low হলে অকার্যকর হয়।
vi. Vo : এটি LCD contrast পিন।

## ৬.২ ইনস্ট্রাকশন রেজিস্টার, ডাটা রেজিস্টার ও বিজি ফ্ল্যাগ বর্ণনা
**(Describe the Instruction Register, Data Register and Busy Flag)**

HD44780 (16 × 2) LCD-এর দুটি রেজিস্টার আছে :

(ক) ডাটা রেজিস্টার (Data Register)
(খ) কমান্ড রেজিস্টার বা ইনস্ট্রাকশন রেজিস্টার (Command Register or Instruction Register)

**(ক) ডাটা রেজিস্টার (Data Register) :** ডাটা রেজিস্টার একটি 8 বিটের রেজিস্টার। এটি LCD-তে প্রদর্শিত তথ্য ধারণ করে। LCD তে প্রদর্শিত তথ্য ASCII মানের হয়। এটি Display data ram (DDRAM) এবং character data RAM (CGRAM)-এ যে তথ্য লিখিত (write) হবে তা সাময়িকভাবে ধারণ করে। DDRAM বা CGRAM হতে যে তথ্য পঠিত (Read) হবে তাও সাময়িকভাবে ধারণ করার জন্য ব্যবহার করা হয়।



Register Select (RS) সিগন্যালের মাধ্যমে উপরের দুই রেজিস্টার নির্বাচন করা হয়। নিচে ছক আকারে দেখানো হলো–

| Register Selector | |
|---|---|
| **RS** | **R/W** |
| 0 | 0 | LCD তে একটি কমান্ড পাঠানো হয়। |
| 0 | 1 | Read *busy flag (DB7) এবং address counter (DB0 to DB6) |
| 1 | 0 | LCD তে তথ্য পাঠায়। |
| 1 | 1 | LCD থেকে তথ্য পড়ে। |

**(খ) কমান্ড রেজিস্টার বা ইনস্ট্রাকশন রেজিস্টার (Command Register or Instruction Register) :** Command register বা Instruction Register একটি 8 বিটের রেজিস্টার। এটি LCD তে প্রদেয় কমান্ড ইনস্ট্রাকশনসমূহ ধারণ করে। LCD তে প্রদেয় একটি নির্দেশনা হলো কমান্ড, যা পূর্বনির্ধারিত কাজ যেমন LCD সূচনা করা, পর্দা পরিষ্কার করা, কার্সরের অবস্থান সেটিং করা, ডিসপ্লে নিয়ন্ত্রণ ইত্যাদি সম্পাদন করে। এটি Display data ram (DDRAM) এবং Character data RAM (CGRAM)-এর তথ্যও ধারণ করে।

**বিজি ফ্ল্যাগ (BF-Busy Flag) :** পূর্বের নির্দেশনা সমাপ্তি বা পরবর্তী নির্দেশনার জন্য প্রস্তুত কি না তার জন্য BF ফ্ল্যাগ LCD তে একটি ইঙ্গিত দেয়।

## ৬.৩ এলসিডি কমান্ড তালিকা
**(List the LCD Command Codes)**

LCD-এর কমান্ড (Hex code)-এর তালিকা নিচে উল্লেখ করা হলো :

| Code (Hex) | LCD Instruction Register Commands |
|---|---|
| 1 | LCD পর্দা পরিষ্কার করে (Clear display screen)। |
| 2 | Return home |
| 4 | কার্সরের অবস্থান বামদিকে একঘর স্থানান্তরিত করে (Shift cursor to left)। |
| 5 | ডিসপ্লেকে ডানদিকে স্থানারিত করে (Shift display right)। |
| 6 | কার্যরের অবস্থান ডানদিকে একঘর স্থানান্তরিত করে (Shift cursor to right)। |
| 7 | ডিসপ্লেকে বামদিকে স্থানান্তরিত করে (Shift display left)। |
| 8 | ডিসপ্লে ও কার্সর অফ (Display off, cursor off)। |
| A | ডিসপ্লে অফ ও কার্সর অন (Display off, cursor on)। |
| C | ডিসপ্লে অন ও কার্সর অফ (Display on, cursor off), |
| E | ডিসপ্লে অন ও কার্সর মিটমিট করে জ্বলে (Display on, cursor blinking)। |
| F | ডিসপ্লে অন ও কার্সর মিটমিট করে জ্বলে (Display on, cursor blinking)। |
| 10 | কার্সরের অবস্থান বামদিকে স্থানান্তরিত করে (Shift cursor position to left)। |
| 14 | কার্সরের অবস্থান ডানদিকে একঘর স্থানান্তরিত করে (Shift cursor position to right)। |
| 18 | সম্পূর্ণ ডিসপ্লেকে বামদিকে স্থানান্তরিত করে (Shift the entire display to the left)। |
| 1C | সম্পূর্ণ ডিসপ্লেকে ডানদিকে স্থানান্তরিত করে (Shift the entire display to the right)। |
| 80 | কার্সরকে ১ম লাইনের শুরুতে আনার জন্য জোর দেয় (Force cursor to beginning to 1st line)। |
| C0 | কার্সরকে ২য় লাইনের শুরুতে আনার জন্য জোর দেয় (Force cursor to beginning to 2nd line)। |
| 38 | 2 লাইন ও 5 × 7 মেট্রিক্স (2 lines and 5 × 7 matrix)। |

CamScanner

## ৬.৪ LCD-তে ডাটা প্রদর্শনের প্রোগ্রাম

**(Write Program for Displaying Data to LCD)**

**'NO' শব্দটি LCD- তে ডিসপ্লে করার জন্য প্রোগ্রাম :**

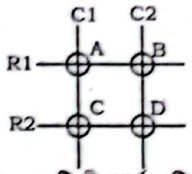
```
; calls a time delay before sending next data/command
; P1.0 - P1.7 are connected to LCD data pins D0 - D7
; P2.0 is connected to RS pin of LCD
; P2.1 is connected to R/W pin of LCD
; P2.2 is connected to E pin of LCD
        ORG 0H
        MOV A, # 38H        ; INIT.LCD 2 LINES, 5×7 MATRIX
        ACALL COMNWRT       ; call command subroutine
        ACALL DELAY         ; give LCD some time
        MOV A, #0EH         ; display on cursor on
        ACALL COMNWRT       ; call command suroutine
        ACALL DELAY         ; give LCD some time
        MOV A, #01          ; clear LCD
        ACALL COMNWRT       ; call command subroutine
        ACALL DELAY         ; give LCD some time
        MOV A, # 'N'        ; display letter N
        ACALL DATA WRT      ; call display subroutine
        ACALL DELAY         ; give LCD some time
        MOV A, # 'O'        ; display letter O
        ACALL DATAWRT       ; call display subroutine
AGAIN : SJMP AGAIN          ; stay here
COMNWRT                     ; send command to LCD
        MOV P1, A           ; copy reg A to port 1
        CLR P2.0            ; RS = 0 for command
        CLR P2.1            ; R/W = 0 for write
        SETB P2.2           ; E = 1 for high pulse
        ACALL DELAY         ; give LCD some time
        CLR P2.2            ; E = 0 for H-to-L pulse
        RET
DATAWRT :                   ; write date to LCD
        MOV P1, A           ; copy reg A to port 1
        SETB P2.0           ; RS = 1 for data
        CLR P2.1            ; R/W = 0 for data
        SETB P2.2           ; E = 1 for high pulse
        ACALL DELAY         ; give LCD some time
        CLR P2.2            ; E = 0 for H-to-L pulse
        RET
DELAY : MOV R3, #50         ; 50 or higher for fast CPUs
HERE2 : MOV R4, #255        ; R4 = 255
HERE :  DJNZ R4, HERE       ; stay until R4 becomes 0
        DJNZ R3, HERE 2
        RET
        END
```



## ৬.৫ একটি মেট্রিক্স কী-বোর্ডের গঠন বর্ণনা

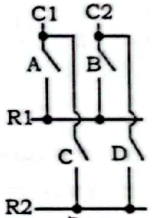
**(Describe the Organization of a Matrix Keyboard)**

মেট্রিক্স কী-বোর্ড খুবই জনপ্রিয়। এর সরল গঠন ও সহজ ইন্টারফেসিং জন্য এটি মাইক্রোকন্ট্রোলারে ব্যাপক ভাবে ব্যবহার করা হয়। কী-বোর্ড একটি HMI (Human Machine Interface)-এর অংশ এবং একটি ছোট embedded system-এ এর ভূমিকা অনেক। মেট্রিক্স কী-বোর্ড তারের তৈরি সারি ও কলামের একটি মেট্রিক্স নিয়ে গঠিত। প্রত্যেক কী একটি সুইচের মতো আচরণ করে। যখন একটি কীতে চাপ প্রয়োগ করা হয় তখন একটি কলামের তার একটি সারির তারের সাথে সংযোগ হয়ে একটি সার্কিট পূর্ণ করে। কী-বোর্ড কন্ট্রোলার এই ক্লোজ সার্কিটকে শনাক্ত করে এবং একটি "কী চাপ" রেজিস্টার করে। পাশে একটি ধারণাসংক্রান্ত সরল কী-বোর্ড মেট্রিক্স সার্কিট দেখানো হলো।



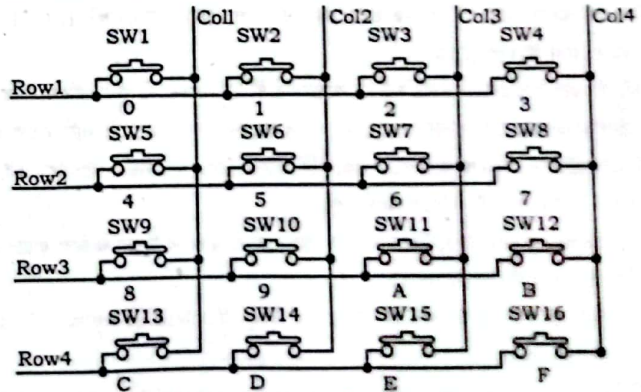

চিত্র-৬.২ : একটি কী-বোর্ড মেট্রিক্স সার্কিট

এই কী-বোর্ডের 4টি কী- A, B, C ও D। গ্রাফের বিন্দুর মতো প্রত্যেক কী এর একটি অদ্বিতীয় গ্রিড লোকেশন আছে। A কী এর নুড C1R1, B কী এর নুড C2R1, C কী এর নুড C1R2 এবং D কী এর নুড C2R2। এর ইলেকট্রনিক সার্কিট দেখতে অনেকটা নিচের চিত্রের মতো।




চিত্র-৬.৩ : সুইচ খোলা

প্রকৃত কী-বোর্ড অনেকগুলো কলাম ও সারি নিয়ে গঠিত। মাইক্রোকন্ট্রোলার পোর্টের মাধ্যমে উভয় সারি ও কলামকে access করে। 8 বিটের পোর্টের সাহায্যে একটি 8 × 8 কী এর মেট্রিক্সকে একটি মাইক্রোকন্ট্রোলারের সাথে সংযোগ করা যায়। যখন একটি কীতে চাপ প্রয়োগ করা হয় তখন একটি সারি ও একটি কলাম একত্রে একটি সংযোগ তৈরি করে, অন্যথায় সারি ও কলামের মাঝখানে কোন সংযোগ হয় না। নিচে 4 × 4 মেট্রিক্সের একটি কী-বোর্ড নিচে দেখানো হলো :




চিত্র-৬.৪ : 8 × 8 মেট্রিক্স কী-বোর্ড