

**Τμήμα Μηχανικών Η/Υ & Πληροφορικής**

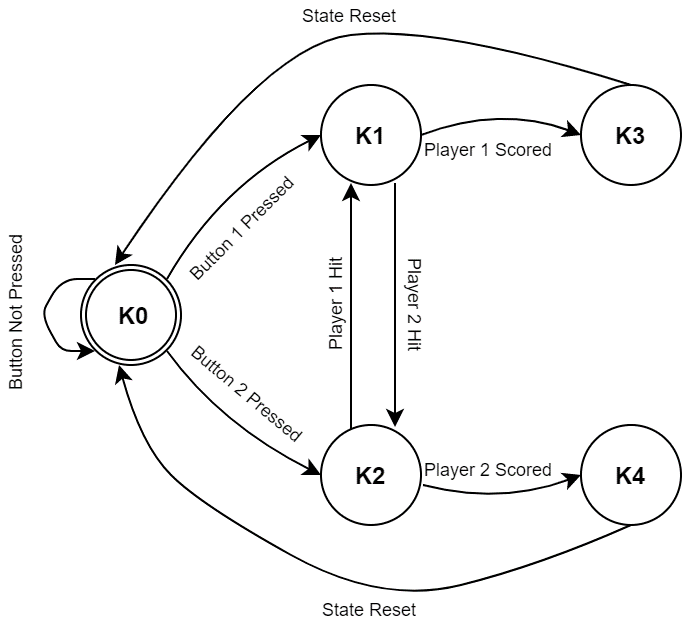
**Εργαστήριο Προηγμένων Μικροεπεξεργαστών**

**Άσκηση 5**

|  |  |
| --- | --- |
| **Ονοματεπώνυμο** | **ΑΜ** |
| Εμμανουήλ Κατεφίδης | 6077 |
| Παναγιώτης Σταυρινάκης | 6217 |

**1. Διάγραμμα Μετάβασης Καταστάσεων**

|  |  |  |  |
| --- | --- | --- | --- |
| **κατάσταση** | **LEDs** | **επόμενη κατάσταση** | **παρατηρήσεις** |
| Κ0(αρχική) | Όλα ΟΝ | εάν πιεστεί το button 1: Κ1 , αλλιώς αν πιεστεί το button 2: K2, αλλιώς: Κ0 | Περιοδική ανίχνευση για πίεση του button 1/2, ο διακόπτης αυτός θέτει ένα flag |
| Κ1 | Πρώτο  ΟΝ | εάν πιεστεί το button 2: Κ2 , αλλιώς: Κ3 | Right-Shift των LEDs |
| Κ2 | Τελευταίο  ΟΝ | εάν πιεστεί το button 1: Κ1 , αλλιώς: Κ4 | Left-Shift των LEDs |
| K3 | Όλα OFF | K0 | Αύξηση του σκορ του πρώτου παίκτη και καθορισμός του (αν είναι) ως νικητή |
| K4 | Όλα OFF | K0 | Αύξηση του σκορ του δεύτερου παίκτη και καθορισμός του (αν είναι) ως νικητή |

****

**2. Κώδικας:**

|  |
| --- |
| #include <sys/types.h>  #include <sys/stat.h>  #include <fcntl.h>  #include <sys/ioctl.h>  #include <unistd.h>  #include <sys/mman.h>  #include <stdio.h>  #include <stdlib.h>  #include <header.h>  #define PIOA\_ID 2  #define TC0\_ID 17  int WhatDoesTheButtonSay(int state);  PIO\* pioa = NULL;  TC\* tc = NULL;  int main( int argc, const char\* argv[] )  {  char tmp;  int state = 0;  int flag = 0;  int i = 0;  int player1\_score = 0;  int player2\_score = 0;  STARTUP; //ΑΡΧΙΚΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ  pioa -> PER = 0x3FF; //ΓΡΑΜΜΗ 1 ΕΩΣ 10: ΓΕΝΙΚΟΥ ΣΚΟΠΟΥ  pioa -> CODR = 0x3FF; //ΓΡΑΜΜΗ 1 ΕΩΣ 10: ΔΥΝΑΜΙΚΟ ΕΞΟΔΟΥ LOW  pioa -> OER = 0x3FF; //ΓΡΑΜΜΗ 1 ΕΩΣ 10: ΛΕΙΤΟΥΡΓΙΑ ΕΞΟΔΟΥ  tc->Channel\_0.RC = 8192; //8192(period of 1 sec)  tc->Channel\_0.CMR = 0x2084; //SLOW CLOCK , WAVEFORM , DISABLE CLK ON RC COMPARE  tc->Channel\_0.IER = 0x10; //ΕΝΕΡΓΟΠΟΙΗΣΗ ΜΟΝΟ ΤΟΥ RC COMPARE  tc->Channel\_0.CCR = 0x05; //ΕΝΕΡΓΟΠΟΙΗΣΗ ΤΟΥ Timer  pioa->PER = 0xC00; //ΓΡΑΜΜΗ 11 ΚΑΙ 12: ΓΕΝΙΚΟΥ ΣΚΟΠΟΥ  pioa -> PUER = 0xC00; //ΕΝΕΡΓΟΠΟΙΗΣΗ ΣΤΗ ΓΡΑΜΜΗ 11 ΚΑΙ 12: PULL−UP  pioa -> ODR = 0xC00; //ΓΡΑΜΜΗ 11 ΚΑΙ 12: ΛΕΙΤΟΥΡΓΙΑ ΕΙΣΟΔΟΥ  while( (tmp = getchar()) != 'e')  {  switch (state)  {  case 0:  pioa->SODR = 0x3FF; //ΕΝΕΡΓΟΠΟΙΗΣΗ ΟΛΩΝ ΤΩΝ LED  state = WhatDoesTheButtonSay(state);  if(state != 0) pioa->CODR = 0x3FF; //ΑΠΕΝΕΡΓΟΠΟΙΗΣΗ ΟΛΩΝ ΤΩΝ //LED  break;  case 1:  pioa->SODR = 0x001; //ΕΝΕΡΓΟΠΟΙΗΣΗ TOΥ ΠΡΩΤΟΥ LED  for(i = 0;i < 9; i++){  pioa->CODR = 0x001<<i;  pioa->SODR = 0x001<<(i+1);  }  flag = WhatDoesTheButtonSay(state);  if( flag == 2 ){  state = 2;  } else {  state = 3;  }  break;  case 2:  pioa->SODR = 0x200; //ΕΝΕΡΓΟΠΟΙΗΣΗ TOΥ ΤΕΛΕΥΤΑΙΟΥ LED  for(i = 0;i < 9; i++){  pioa->CODR = 0x200>>i;  pioa->SODR = 0x200>>(i+1);  }  flag = WhatDoesTheButtonSay(state);  if( flag == 1 ){  state = 1;  } else {  state = 4;  }  break;  case 3:  pioa->CODR = 0x3FF; //ΑΠΕΝΕΡΓΟΠΟΙΗΣΗ ΟΛΩΝ ΤΩΝ LED  player1\_score++;  if( player1\_score == 3 ){  printf("-------------------------\nResult --> Player 1 Wins!\n Insert (bit)Coin..\n-------------------------\n");  player1\_score = 0;  player2\_score = 0;  }  state = 0;  break;  case 4:  pioa->CODR = 0x3FF; //ΑΠΕΝΕΡΓΟΠΟΙΗΣΗ ΟΛΩΝ ΤΩΝ LED  player2\_score++;  if( player2\_score == 3 ){  printf("-------------------------\nResult --> Player 2 Wins!\n Insert (bit)Coin..\n-------------------------\n");  player1\_score = 0;  player2\_score = 0;  }  state = 0;  break;  }  }  tc->Channel\_0.CCR = 0x02; //ΑΠΕΝΕΡΓΟΠΟΙΗΣΗ ΤΟΥ Timer  CLEANUP;  return 0;  }  int WhatDoesTheButtonSay(int state)  {  unsigned int data\_in = 0;  int flag = 0;    data\_in = pioa->PDSR; //ΑΝΑΓΝΩΣΗ ΤΙΜΩΝ ΕΙΣΟΔΟΥ    if( (data\_in & 0x800) && ( state == 0 || state == 2) ) {//ΔΙΑΚΟΠΤΗΣ Player 1 //ΠΑΤΗΜΕΝΟΣ(1000 0000 0000(b))  flag = 1;  }else if ( (data\_in & 0x400) && ( state == 0 || state == 1) ) { //ΔΙΑΚΟΠΤΗΣ //Player 2 ΠΑΤΗΜΕΝΟΣ(0100 0000 0000(b))  flag = 2;  }  return flag;  } |