```
Name Edison_main;
Assembly 0001;
Revision 10.0;
PartNo U4 ATF1508AS;
Device f1508ispplcc84;
Company S100Computers.com;
Designer John Monahan;
Location CA,San Ramon;
Date 15 March 2017;


property ATMEL { xor_synthesis=on };
property ATMEL { logic_doubling=on };
property ATMEL { jtag=on };
PROPERTY ATMEL { preassign keep };
PROPERTY ATMEL { TMS_pullup=on };
PROPERTY ATMEL { TDI_pullup=on };


/*
* -----------------  Edison BOARD CPLD IS SETUP TO RUN AS A S100 BUS SLAVE DEVICE (V0.6) ------------------------
* Pin assignments assuming V1.2 (Most signals are active low)
* Make all data and address outputs fast slew and all chip selects slow
* For V1.21 Board:-

* Modifications:-
* V1.21a             ;Switched the polarity of the Stop Switch signal. Now HIGH going to the Edison halts everything

*/

Pin 83 = MASTER_CLK;                    /* FAST Oscillator (Start with 5 MHz or from Edison JP17 pin 13*/
Pin 2  = LOCAL_PHI;                     /* Phi Input from S100 bus */
Pin 81 = S100_CLK;

Pin 4  = EDISON_READY_LED;              /* Active low */
Pin 5  = DIAG_LED2;
Pin 6  = DATA_OUT_LED;
Pin 8  = DATA_IN_LED;

Pin 9  = IO_RAM_WR;
Pin 10 = IO_RAM_RD;
Pin 11 = CPLD_RESET;
Pin 15 = E_STOP_REQUEST;                /* GP135, A HIGH stops the Edison in its tracks */

Pin 16 = E_sINP;                        /* GP128  Control lines to CPLD from Edison (U7) */
Pin 17 = E_sOUT;                        /* GP13_PWM1 */
Pin 18 = E_MEMR;                        /* GP165 */
Pin 20 = E_MEMW;                        /* GP19 */
Pin 21 = P16;                           /* Spare, GP12_PWM0 */
Pin 22 = DATA_WR;                       /* GP183_PWM3 */

Pin 24 = EDISON_READY;                  /* GP110 */
Pin 25 = DATA_RD;                       /* GP114 */
```

```
Pin 27 = ACTIVATE_EDISON;                /* GP135 */
Pin 28 = P21;                            /* Spare */
Pin 29 = RW_PULSE;                       /* GP49 */
Pin 30 = STOP_SWITCH;                    /* SW1 High to Low to pulse to Edison to interrupt/stop current process */
Pin 31 = P20;                           /* Spare */
Pin 33 = E_pSYNC;                        /* GP83 */
Pin 34 = E_sINTA;                        /* GP48 */

Pin 35 = TMAx;
Pin 36 = XFERII;                         /* Active LOW */
Pin 37 = XFERI;                          /* Active LOW */
Pin 39 = INT_REQ;
Pin 40 = NMI_REQ;

Pin 41 = pHLDA;                          /* S100 bus Hold Acknowledge */
Pin 44 = HOLD;                           /* S100 bus Hold request Active Low */

Pin 46 = ADDRESS1;                       /* Latch Address lines 0-7 */
Pin 48 = ADDRESS2;                       /* Latch Address lines 8-15 */
Pin 49 = ADDRESS3;                       /* Latch Address lines 16-23 */

Pin 50 = E_RESET_CMD;
Pin 51 = SLAVE_RESET;
Pin 52 = E_S100_INT;                     /* S100 Interrupt from S100 bus to Edison U23,B7 */
Pin 54 = P15_JUMPERS;
Pin 55 = LATCH_ADD3;                     /* latch U8 */
Pin 56 = LATCH_ADD2;                     /* latch U9 */
Pin 57 = LATCH_ADD1;                     /* latch U2 */


Pin 58 = S100_INT_VECTOR;
Pin 60 = S100_PHANTOM;                   /* Active Low */

Pin 61 = S100_DATA_IN;                   /* Active LOW */
Pin 63 = S100_DATA_OUT;                  /* Active LOW */
Pin 64 = bMEM_RD;
Pin 65 = bsINP;
Pin 67 = bsOUT;
Pin 68 = bsM1;
Pin 69 = bsWO;
Pin 70 = bINTA;
Pin 73 = bHALT;
Pin 74 = bMEM_WR;

Pin 75 = bpWR;                           /* Active LOW */
Pin 76 = bpDBIN;                         /* Active HIGH */

Pin 77 = bsXTRQ;                         /* Active Low */
Pin 79 = bpSYNC;
Pin 80 = bpSTVAL;

Pin 84 = NC4;
Pin 1  = MASTER_RESET;                   /* S100 Bus reset. Active LOW */
```

```
Pin 12 = INACTIVATE_DATA_LINES;           /* S100 bus control of status lines */
Pin 45 = INACTIVATE_CONTROL_LINES;        /* S100 bus control of data, address and control lines */


/* ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////// */

Pinnode =  [reg3..0];                                          /* Need a few D typer Flip Flops below*/
Pinnode =  BOARD_ACTIVE;


!EDISON_READY_LED    =  !EDISON_READY;                         /* This line is HIGH only after the Edison has initilized Linux */

reg0.d = TMAx & !EDISON_READY;                                 /* TMAx goes low-> high */
reg0.ckmux = !LOCAL_PHI;
reg0.ar = !MASTER_RESET;                                       /* Return to high on reset */
!HOLD   = reg0 & MASTER_RESET;                                 /* Lower S100 bus HOLD line WHEN the Edison has completed loading Linux */


!XFERI = reg0 & pHLDA & MASTER_RESET & !EDISON_READY;          /* Only activate when Edison is ready */

reg1.d  = reg0;
reg1.ck = pHLDA;
reg1.ar = !reg0;
!XFERII = reg1;

BOARD_ACTIVE = reg1 & MASTER_RESET & !EDISON_READY;            /* Active HIGH */
                                                               /* Note this will also light up the Edison Active LED */
reg2.d  = reg1;
reg2.ckmux = !LOCAL_PHI;
reg2.ar = !reg1;

!ACTIVATE_EDISON  = reg2 & MASTER_RESET & BOARD_ACTIVE & !EDISON_READY;      /* LOWER the Activate Edison line (U12, pin 19) */

Pinnode = [CD3..0];
                                                                        /* MASTER_CLK = 16 MHz */
CD0.t = 'b'1;                                                           /* 8 MHz */
CD1.t = CD0;                                                            /* 4 MHz */
CD2.t = CD0 & CD1;                                                      /* 2 MHz */
CD3.t = CD0 & CD1 & CD2;                                                /* 1 MHz */

[CD3..0].ckmux = MASTER_CLK;


S100_CLK  = CD0;                                                        /* Run Phi at 8Hz for now */

INACTIVATE_DATA_LINES =  'b'1;                                          /* Just to be safe, for now,  pull these unimplemented signals high */
INACTIVATE_CONTROL_LINES =  'b'1;
CPU_HOLD     =  'b'0;                                                   /* For now ignore */
bsXTRQ   = 'b'1;                                                        /* Always HIGH, for this 8 bit board */


/* Synthesize the S100 Bus Control and Status signals  */
```

```
                                                                    /* ====== ADDRESS BUFFERS CONTROL ================== */
LATCH_ADD1           = !ADDRESS1  & !XFERII;                        /* Latch U8 data on high */
LATCH_ADD2           = !ADDRESS2  & !XFERII;                        /* Latch U9 data on high */
LATCH_ADD3           = !ADDRESS3  & !XFERII;                        /* Latch U2 data on high */


                                                                    /* ====== S100 BUS DATA FROM EDISON (U4)  ========== */
!S100_DATA_OUT= !DATA_WR &  !XFERII;                                /* Data OUT via U4 OE = LOW */
IO_RAM_WR            = (!E_MEMW  # !E_sOUT # !E_sINTA) & !S100_DATA_OUT;            /* Raise pin 11 of U4 to LATCH DATA. Note, active HIGH */


                                                                /* ====== S100 BUS DATA TO EDISON (U5)  ============== */
!S100_DATA_IN        = !DATA_RD  & !XFERII;                     /* Data IN via U5 OE = LOW */
IO_RAM_RD            = (!E_MEMR  # !E_sINP) & !S100_DATA_IN;    /* Raise pin 11 of U5 to LATCH DATA. Note, active HIGH */

                                                                /* ============ STATUS LINES ========== */
bpSYNC        = !E_pSYNC & !XFERII;
!bpSTVAL             =  bpSYNC  & !XFERII;                      /* Pulse S100 bpSTVAL line, LOW */

!bsOUT        = !E_sOUT  & !XFERII;                             /* sOUT status signal to S100 bus, HIGH, inverted by U16 */

!bsINP        = !E_sINP & !XFERII;                              /* sINP status signal to S100 bus, LOW here , but inverted by U16 to HIGH */

!bMEM_WR             = !E_MEMW & !XFERII;                       /* MWRT status signal to S100 bus, LOW here , but inverted by U16 to HIGH */

!bMEM_RD             = !E_MEMR & !XFERII;                       /* sMEMR status signal to S100 bus, LOW here , but inverted by U16 to HIGH */

!HALT                = !E_STOP_REQUEST & !XFERII;               /*  S100 bus "HALT" line (is 8080 Specific -- not really used on S100 bus */

!bsM1                = bpSYNC # !bINTA ;                        /* CPU code read for S100 bus. Not really used on S100 bus */

bsWO                 = IO_RAM_WR & !XFERII;                     /* I/O Write */


                                                                /* ============ CONTROL LINES ========== */

bpDBIN        = !DATA_RD & !RW_PULSE  & !XFERII;                /* Command to pulse S100 bpDBIN line, HIGH */

!bpWR                = !DATA_WR & !RW_PULSE  & !XFERII;         /* Command to pulse S100 bus bpWR* line, LOW */

!bINTA        = !E_sINTA & !RW_PULSE  & !XFERII;                /* Pulse S100 bus sINTA line (High on S100 bus via U16) */

!SLAVE_RESET         = !E_RESET_CMD & BOARD_ACTIVE;             /* Return control back to S100 master (Z80) */

!E_S100_INT          = S100_INT_VECTOR & !XFERII;               /* Interrupt from S100 bus */


                                                                /* ====== LEDs etc. ========== */
!DATA_OUT_LED        = !S100_DATA_OUT;                          /* Flash Data Direction LEDs */
!DATA_IN_LED         = !S100_DATA_IN;

E_STOP_REQUEST      = !STOP_SWITCH & (!bsOUT # !bsINP # !bMEM_WR #!bMEM_RD # !bINTA) & !EDISON_READY;  /*  HIGH with Switch SW1 grounded */
```

```
!DIAG_LED2            =  E_STOP_REQUEST;
```