

```
Name Edison_main;
Assembly 0001;
Revision 10.0;
PartNo U4 ATF1508AS;
Device f1508ispplcc84;
Company S100Computers.com;
Designer John Monahan;
Location CA, San Ramon;
Date 5/7/2017;
```

```
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 };
```

```
/*
 * ----- EdisonII BOARD CPLD IS SETUP TO RUN AS A S100 BUS SLAVE DEVICE (V0.6) -----
 * Pin assignments assuming V2.0 Board (Most signals are active low)
 * Make all data and address outputs fast slew and all chip selects slow
 */

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_LED1;
Pin 6  = BUS_DI_READ;
Pin 8  = BUS_DO_READ;

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 = BUS_STATUS_READ;    /* 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 = BUS_CTL_READ;
Pin 29 = RW_PULSE;           /* GP49 */
```

```

Pin 30 = SS_SWITCH;           /* SW1 High to Low to pulse to Edison to interrupt/stop current process */
Pin 31 = TO_BUS;             /* GP131 From Edison, LOW if Edison controls S100 bus, HIGH if Edison monitors bus*/
Pin 33 = E_pSYNC;           /* GP83 */
Pin 34 = E_sINTA;           /* GP48 */

Pin 35 = TMAx;
Pin 36 = XFERI1;           /* Active LOW */
Pin 37 = XFERI;            /* Active LOW */
Pin 39 = BUS_ADD1_READ;
Pin 40 = BUS_ADD2_READ;

Pin 41 = BUS_ADD3_READ;    /* 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_WAIT;           /* GP79 */
Pin 51 = SLAVE_RESET;
Pin 52 = S100_INT;        /* S100 Interrupt from S100 bus */
Pin 54 = S100_XRDY;        /* Put wait states on the S100 bus when in monitoring mode */
Pin 55 = LATCH_ADD3;       /* latch U8 */
Pin 56 = LATCH_ADD2;       /* latch U9 */
Pin 57 = LATCH_ADD1;       /* latch U2 */

Pin 58 = E_S100_INT;       /* Send Interrupt to Edison U23,B7*/
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 = pHLDA;

Pin 84 = NC1;
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 */

```



```

LATCH_ADD1      =  ((!ADDRESS1  & !XFERII & !TO_BUS)
high) */
                #  (!ADDRESS1  & XFERII & TO_BUS));

!DIAG_LED1      =  LATCH_ADD1;

LATCH_ADD2      =  !ADDRESS2  & !XFERII & !TO_BUS;
LATCH_ADD3      =  !ADDRESS3  & !XFERII & !TO_BUS;

!BUS_ADD1_READ  =  !ADDRESS2  & ADDRESS3 & !RW_PULSE & TO_BUS;
!BUS_ADD2_READ  =  ADDRESS2  & !ADDRESS3 & !RW_PULSE & TO_BUS;
!BUS_ADD3_READ  =  !ADDRESS2  & !ADDRESS3 & !RW_PULSE & TO_BUS;

!S100_DATA_OUT= !DATA_WR & !XFERII;
IO_RAM_WR       =  (!E_MEMW  # !E_sOUT # !E_sINTA) & !S100_DATA_OUT;

!S100_DATA_IN   =  !DATA_RD  & !XFERII & !TO_BUS;
IO_RAM_RD       =  (((!E_MEMR  # !E_sINP) & !S100_DATA_IN & !TO_BUS)
                #  (!E_MEMR  & !RW_PULSE & XFERII & TO_BUS));

bpSYNC          =  !E_pSYNC & !XFERII;
!bpSTVAL        =  bpSYNC & !XFERII;

!bsOUT          =  !E_sOUT  & !XFERII;

!bsINP         =  !E_sINP  & !XFERII;

!bMEM_WR        =  !E_MEMW  & !XFERII;
!bMEM_RD        =  !E_MEMR  & !XFERII;

!HALT           =  !E_STOP_REQUEST & !XFERII;

!bsM1           =  bpSYNC # !bINTA ;

bsWO            =  IO_RAM_WR & !XFERII;

bpDBIN          =  !DATA_RD & !RW_PULSE & !XFERII;
!bpWR           =  !DATA_WR & !RW_PULSE & !XFERII;
!bINTA          =  !E_sINTA & !RW_PULSE & !XFERII;

/* ===== ADDRESS BUFFERS CONTROL ===== */
/* Latch U8 data on high (and U32 + U33 +U34 if TO_BUS is
high) */
/* Latch U9 data on high */
/* Latch U2 data on high */
/* For bus monitoring OE* lowest 8 bit address on U32 */
/* For bus monitoring OE* middle 8 bit address on U33 */
/* For bus monitoring OE* highest 8 bit address on U34 */
/* ===== S100 BUS DATA FROM EDISON (U4) ===== */
/* Data OUT via U4 OE = LOW */
/* Raise pin 11 of U4 to LATCH DATA. Note, active HIGH */
/* ===== S100 BUS DATA TO EDISON (U5) ===== */
/* Data IN via U5 OE = LOW */
/* Raise pin 11 of U5 to LATCH DATA. Note, active HIGH */
/* OR, when in monitoring mode to latch data for U22 & U35 */

/* ===== STATUS LINES ===== */
/* Pulse S100 bpSTVAL line, LOW */
/* sOUT status signal to S100 bus, HIGH, inverted by U16 */
/* sINP status signal to S100 bus, LOW here , but inverted by U16 to HIGH */
/* MWRT status signal to S100 bus, LOW here , but inverted by U16 to HIGH */
/* sMEMR status signal to S100 bus, LOW here , but inverted by U16 to HIGH */
/* S100 bus "HALT" line (is 8080 Specific -- not really used on S100 bus */
/* CPU code read for S100 bus. Not really used on S100 bus */
/* I/O Write */

/* ===== CONTROL LINES ===== */
/* Command to pulse S100 bpDBIN line, HIGH */
/* Command to pulse S100 bus bpWR* line, LOW */
/* Pulse S100 bus sINTA line (High on S100 bus via U16) */

```

```

!SLAVE_RESET      = !E_WAIT & BOARD_ACTIVE & !TO_BUS;      /* Return control back to S100 master (Z80) IF Edison controls bus*/
!E_S100_INT       = S100_INT & !XFERII;                    /* Interrupt from S100 bus */
!S100_XRDY        = !E_WAIT & BOARD_ACTIVE & TO_BUS;       /* Add (after continous) wait states to S100 bus if in monitoring mode */

/* ===== S100 BUS DATA TO EDISON (Monitoring mode >> ONLY <<) ===== */
/* These signals will NEVER see the S100 bus because XFERII & TO_BUS are high
*/
/* remember also they are counting on IO_RAM_RD from above pulsing high */

!BUS_STATUS_READ = !E_sOUT & !E_WAIT & XFERII & !RW_PULSE & TO_BUS; /* For monitoring Mode use E_sOUT signal for OE* for U7 */
!BUS_CTRL_READ   = !E_sINP & !E_WAIT & XFERII & !RW_PULSE & TO_BUS; /* For monitoring Mode use E_sINP signal for OE* for U31 */
!BUS_DI_READ     = !E_sINTA & !E_WAIT & XFERII & !RW_PULSE & TO_BUS; /* For monitoring Mode use E_sINTA signal for OE* for U22 */
!BUS_DO_READ     = !E_pSYNC & !E_WAIT & XFERII & !RW_PULSE & TO_BUS; /* For monitoring Mode use E_pSYNC signal for OE* for U35 */

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

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

```