EE 308 – LAB 11

Memory Expansion for the HC12

In this lab you will will add external memory to your HC12. You will be given a daughter board which will plug into the header pins on your HC12 EVBU. The daughter board has an Altera EPM7128S chip which will function as the address latch and memory decoder needed for the memory expansion. This week you will program the Altera chip to enable the memory expansion. Next week you will add code to the Altera chip to add two general purpose I/O ports.

The attached schematic shows how the Altera chip and RAM chip are connected. Note that the lines between the Altera chip and the HC12 are:

AD15-0	Ports A and B
$\overline{\text{IRQ}}$	Port E1
R/$\overline{\text{W}}$	Port E2
$\overline{\text{LSTRB}}$	Port E3
E	Port E4

The lines connected to the RAM chip are:

AD15-0
A16-0
$\overline{\text{CS\_MEM}}$
$\overline{\text{WE}}$
$\overline{\text{OE}}$
$\overline{\text{LSTRB}}$

Note that the the RAM chip uses 16 address lines to select one of 2^{16} data words (64 kW), while the HC12 uses 16 address lines to select one of 2^{16} data bytes (64 kB). The RAM chip uses its $\overline{\text{BHE}}$ (High Byte Enable) and $\overline{\text{BLE}}$ (Low Byte Enable) lines to distinguish between bytes within a data word. Thus, the address lines for the HC12 and the RAM do not quite match. It is necessary to connect address line A1 from the HC12 to address line A0 of the RAM; A2 from the HC12 to A1 of the RAM, etc. Use A0 from the HC12 to $\overline{\text{BHE}}$ of the RAM to select the high (even) byte. Use $\overline{\text{LSTRB}}$ from the HC12 to $\overline{\text{BLE}}$ of the RAM to select the low (odd) byte.

Note that on the schematic that there is something called the JTAG interface. This is simply some circuitry which will allow you to program the Altera chip through the parallel port of your computer.

1. Complete the attached Altera GDF file to finish the memory expansion. (You can do it as a TDF file if you prefer.) About all you need to do is to assign the pinouts of the Altera chip to match the pinouts shown in the schematic. We are not going to have the Altera chip interrupt the HC12 at this time, so you should add to the Altera file something which will drive the $\overline{\text{IRQ}}$ line out of the HC12 high. Also, you have to figure out what you should do with the A16 line out of the Altera chip (which goes to the A15 line of the RAM).

   The port expansion next week will require the use of the RESET line from the HC12, which will be connected to Pin 89 of the Altera chip. You should add a RESET input pin to the GDF (or TDF) file, and assign it to Pin 89 to reserve this pin.

2. Compile your Altera program and download it into the Altera chip on your memory expansion board. Your lab instructor will show you how to do this. (Also, have your lab instructor check your Altera program and pinouts. It is possible to damage the Altera chip, the RAM chip, and/or the HC12 if the pin assignments are incorrect.)

3. Put the HC12 into expanded mode. To do this, put the following program into EEPROM at address 0x0D00:

```
CLR   $0016       ; Disable COP
MOVB  #$2C,$000A  ; Enable LSTRB, R/W and E
BSET  $000B,#$68  ; Expanded wide mode, turn on internal visibility
BCLR  $0013,#$0C  ; Put in zero E-Clock stretches
JMP   $F700       ; Jump to DBug-12
```

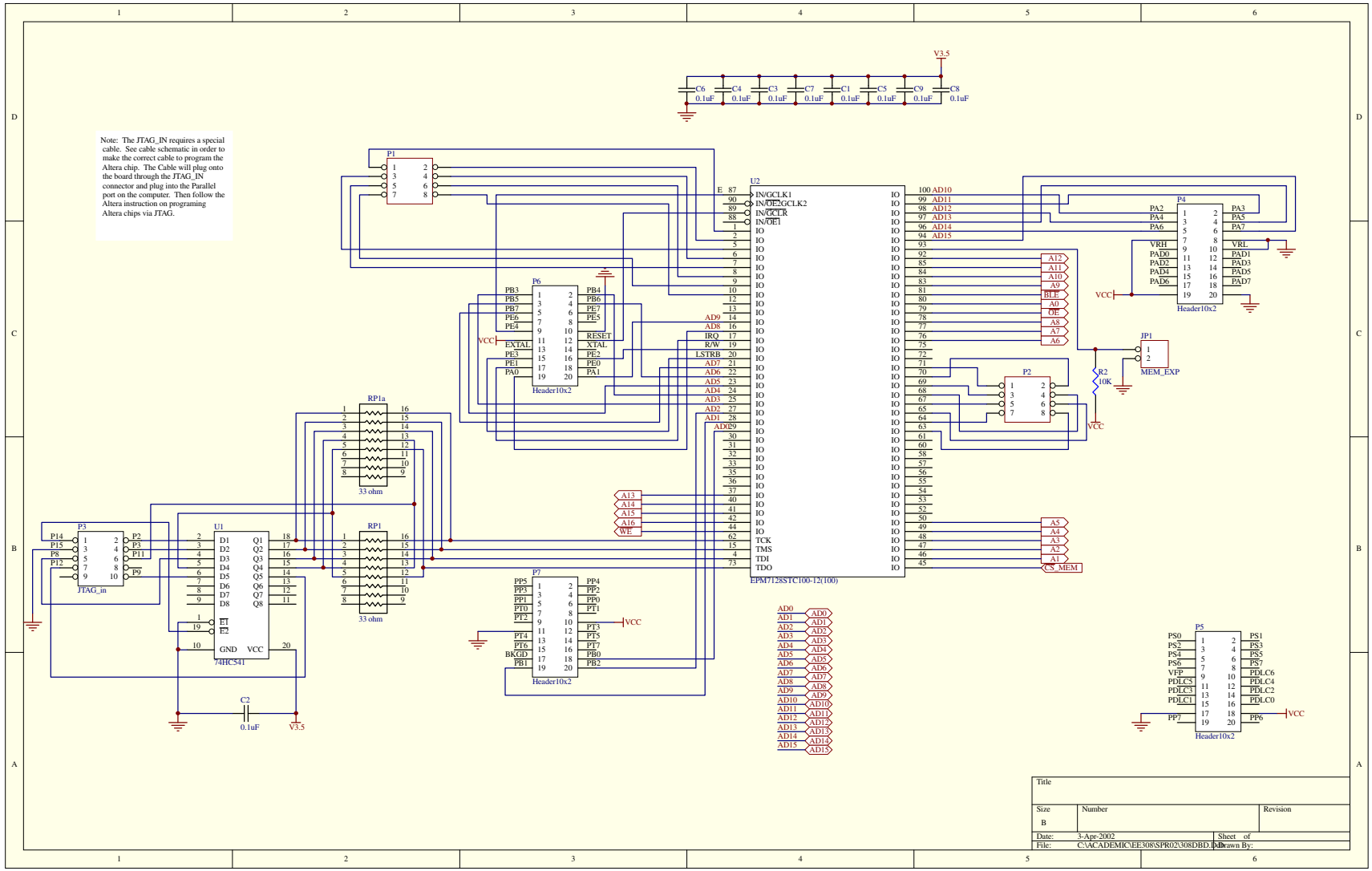
   Move the jumpers so the HC12 will run from EEPROM. Reset the HC12. Verify you are in expanded mode by using DBug-12 to examine the contents of the PEAR, MODE and MISC registers. Record the values of these three registers, and show that the HC12 is in expanded mode, that LSTRB, R/W and E are enabled, and that there are zero E-clock stretches.

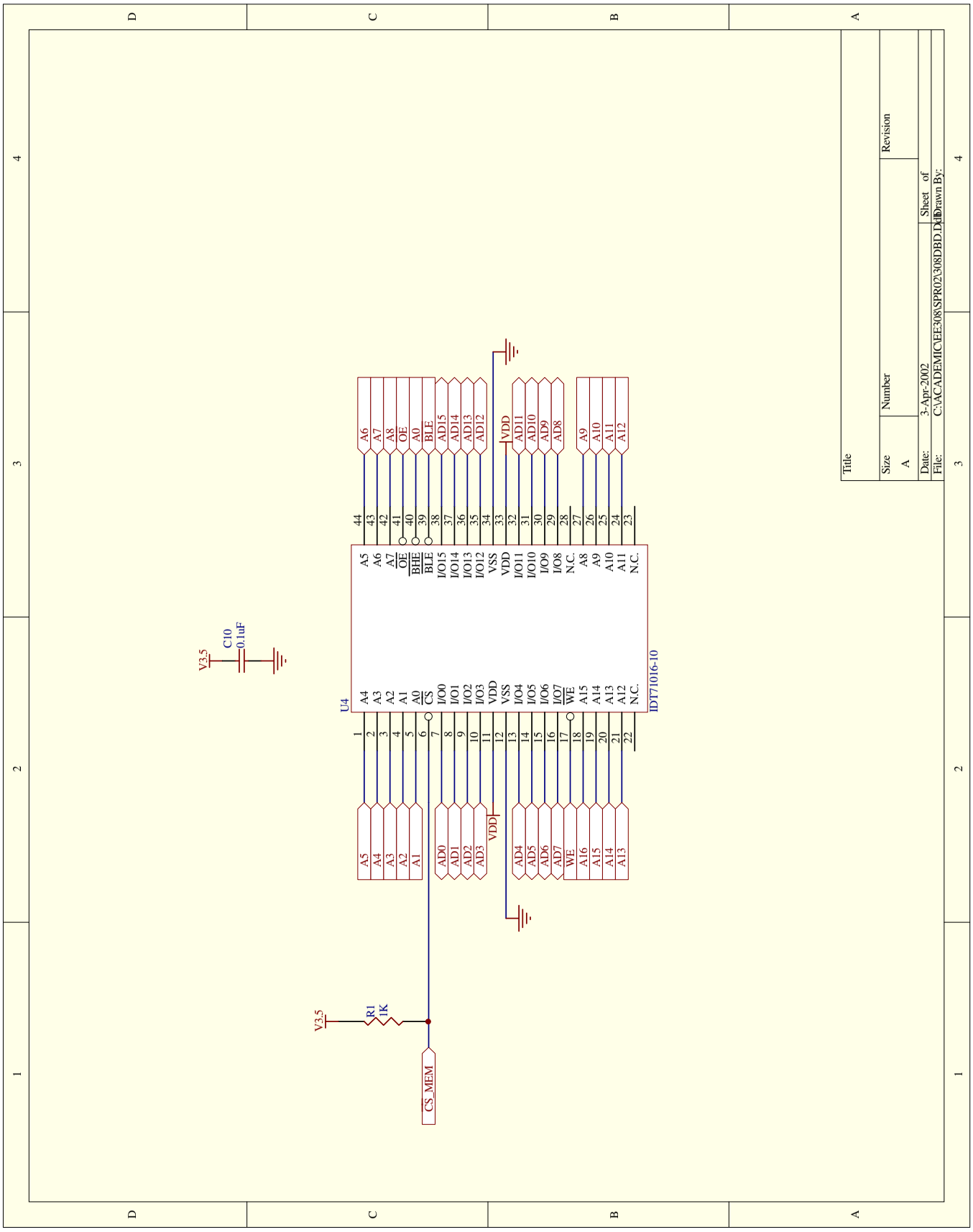
4. Verify that you can access the expanded memory. Use DBug-12 to do a Block Fill of memory from 0x1000 to 0x7FFF. First fill it with 0x55, then 0xAA (alternating patterns of 0 and 1). If this works, your memory expansion is probably okay.

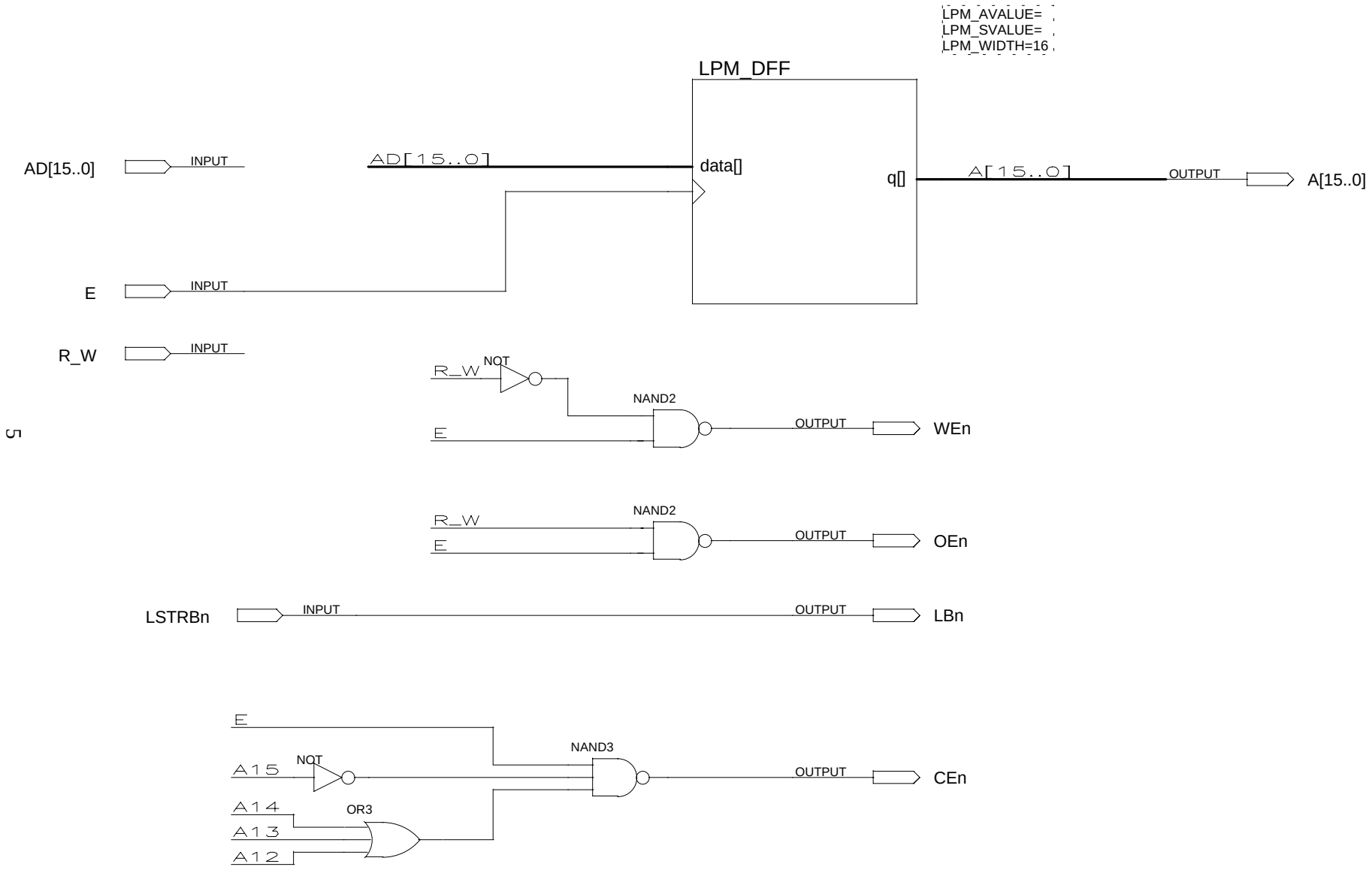
5. Take your HC12 to a logic analyzer. Connect the following lines to the logic analyzer: AD3-0, AD15-12, LSTRB, R/W, and E. (There are not enough pins on the logic analyzer to connect all the address/data and control lines from the HC12). Set up the logic analyzer to use the fastest clock possible. Use DBug-12 to enter the following simple program into memory starting at address 0x2000:

```
movw  #$aa55,$0000
movb  #$cc,$0002
movb  #$dd,$0003
bra   $2000
```

   This program will do a 16-bit access of an even address, an 8-bit access of an even address, and an 8-bit access of an odd address. Capture a waveform while this program is running. Show that an E-clock cycle takes about 125 ns. Identify the three types of memory access, and verify that LSTRB, A0, and R/W are what they should be. Also, verify that the Address/Data lines display the address when E is low, and data when E is high. (Note: The results may be a bit ambiguous because of the time it takes for AD15-0 to switch from data to address after E goes low, and the time it takes for AD15-0 to switch from address to data after E goes high.) The logic analyzer is not fast enough to allow you to make accurate timing measurements to compare to the specifications in the data sheet, but should be fast enough to show the functioning of the address/data and control buses.


Note: The JTAG_IN requires a special cable. See cable schematic in order to make the correct cable to program the Altera chip. The Cable will plug onto the board through the JTAG_IN connector and plug into the Parallel port on the computer. Then follow the Altera instruction on programing Altera chips via JTAG.
V3.5
C6 0.1uF
C4 0.1uF
C3 0.1uF
C7 0.1uF
C1 0.1uF
C5 0.1uF
C9 0.1uF
C8 0.1uF
P1
U2
EPM7128STC100-12(100)
P4
Header10x2
P6
Header10x2
P7
Header10x2
P5
Header10x2
P2
P3
JTAG_in
U1
74HC541
C2
0.1uF
RP1a
RP1
33 ohm
33 ohm
JP1
MEM_EXP
R2
10K
VCC
Title
Size B
Number
Revision
Date: 3-Apr-2002
Sheet of
File: C:\ACADEMIC\EE308\SPR02\308DBD.IndDrawn By:


U4
IDT71016-10
C10
0.1uF
V3.5
R1
1K
CS_MEM
VDD
Title
Size
A
Number
Revision
Date: 3-Apr-2002
Sheet of
File: C:\ACADEMIC\EE308\SPR02\308DBD.DDB
Drawn By:


LPM_AVALUE=
LPM_SVALUE=
LPM_WIDTH=16
LPM_DFF
AD[15..0]
INPUT
AD[15..0]
data[]
q[]
A[15..0]
OUTPUT
A[15..0]
E
INPUT
R_W
INPUT
R_W
NOT
NAND2
E
OUTPUT
WEn
R_W
NAND2
E
OUTPUT
OEn
LSTRBn
INPUT
OUTPUT
LBn
E
A15
NOT
NAND3
A14
A13
A12
OR3
OUTPUT
CEn