UF 68HC12 Development Kit (Board Version 4.2) Manual

Contents

1.0 Introduction

The UF 68HC12 Development Kit integrates the 68HC12B32 with an EEPROM/EPROM and an Altera 7064 CPLD. The purpose of the EEPROM/EPROM is to allow expanded mode booting of the 6812 using the UF-6812 Monitor/Debugger, D-Bug4744 (similar to Motorola's D-Bug12). I/O can be easily connected to the processor via address decoding in the CPLD. These features, coupled with the fact that the 6812's address, data, control and port pins are broken out to 100 mil headers, provide a fast and easy platform for prototyping of hardware with the 68HC12B32 microcontroller. A USB port enables easy downloading of programs from any PC. A JTAG header enables easy reprogramming of the CPLD. A large prototyping area is provided for easy expansion (using wirewrap and soldering) to the provided headers locations along top of the prototyping area. Power and ground pin header locations are provided along both sides of the prototyping area.

The first version of the *UF 68HC12 Development Kit* was designed by Scott Kanowitz under the supervision of Dr. Schwartz and Dr. Gugel. Patrick O'Malley was the originally designer of the *UF D-Bug 4744 Monitor* program. William Goh was the designer of the version 4.0 board under the supervision of Dr. Gugel and Dr. Schwartz. Others working on the monitor program were Robert Oaks, Robert Hill, Dr. Schwartz, Jate Sujjavanich and Adam Barnet.

2.0 Installation & Quick Testing

The following two sections will help you boot up your UF 6812 Development Board.

2.1 Installing the USB Drivers

Install the USB driver from http://www.ftdichip.com/Drivers/CDM/CDM 2.02.06.exe . The driver is also available on our website at http://mil.ufl.edu/4744/software/USB_Drivers/FTDI_FT232RL/ .

2.1.1 USB Driver Conflicts:

When both the USB-Blaster and the USB programming cable are plugged into a board at the same time there may be a driver conflict between the USB FTDI and Altera Byte Blaster drivers. This will cause the Mini-IDE terminal (or other terminal program) to freeze and prevent Quartus from successfully programming the CPLD. You can solve this problem by changing the name of the FTDIBUS.sys driver to xFTDIBUS.sys. You will also have to edit the usbblst.inf file and change all references to FTDIBUS.sys to x FTDIBUS.sys and then update the driver. These files are probably located in C:\altera\*\quartus\drivers\usb-blaster\x??, where ??a depends on the version of Quartus you installed and may be 32 or 64, depending on the processor your computer uses. (It would not hurt to modify the files both the x32 and the x64 folders.)

Some boards may not be assigned to the correct COM port when plugged into a computer, even when the same USB drive is used each time. The easiest fix for this is to see which COM port is assigned to a board that does not have this issue, and then use MiniIDE (or other terminal program) to manually reassign the COM port for the bad board. If another board is unavailable, you will have to test your board with every COM port until you find the one that works.

The above problems are a result of both the USB-Blaster and our PCB using FTDI chips.

2.2 Booting your UF 68HC12 Development Board

1. Open a terminal application on your PC. (I use either HyperTerminal or a MiniIDE Terminal window.) Set your terminal properties to:

Connect Using:	Communication Port Assigned to USB Port (e.g., Com3)
Transfer Rate:	9600 bits per second
Data Bits:	8
Parity:	None
Stop Bits:	1
Flow Control:	None

2. Press the reset button on the board.
3. You should now see the monitor menu show up in your terminal.
4. If you do not see this menu in your terminal application, check all of your board’s jumper settings using the definitions in the next section. Also verify that you have a wirewrap connection between PAD7and ground. (Pin 7 on the A/D port, PAD7, is pin 2 on J4.)

3.0 Header Definitions

This section provides a detailed description of the headers on the board as well as their functions. See Figure 1 for header location. (Note that the bubbles on the header pins on the schematics do **not** represent activation-level information. Headers are passive devices.)

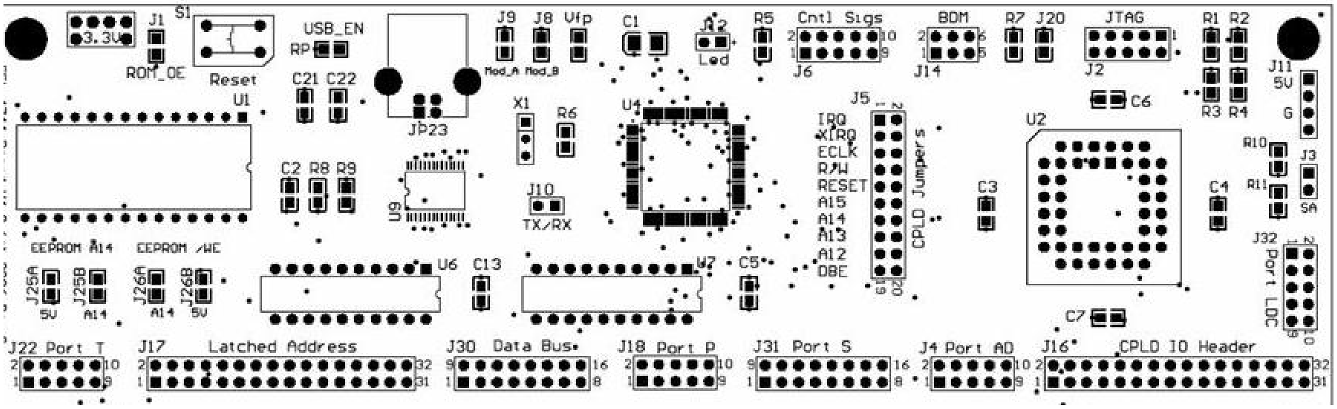


Figure 1: Header and jumper definitions.

J15 (+3.3V) – +3.3V Header. The 3.3V is generated from the FT232RL IC chip. (On top left of the board and labeled 3.3V.)

S1 (Reset) – Reset switch.

JP23 (USB) – Reset switch connector.

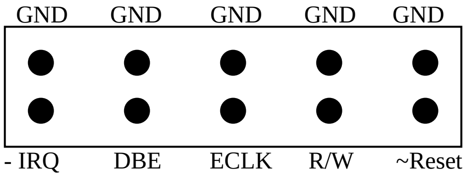


Figure 2: Header J6 Pin Definitions

J6 (Cntl Sigs) – J6 contains some commonly used bus control signals. The pin-out of this header is shown in Figure 2. (- denotes an active low signal)

J14 (BDM) – J14 is the Background Debug Mode (BDM) header from the HC12. It is the standard BDM header as defined by Motorola/Freescale.

J2 (JTAG) – J2 is the JTAG header for the CPLD. This is the standard JTAG programming header.

J11 (5V) – J11 is the input for a regulated 5V power source. Note: To use this as a power input, make sure that jumper RP (USB_EN) is **NOT** installed. This is an alternative power input; generally we do **NOT** use this input and instead get power from the USB port.

J10 (RX/TX) – J10 allows the user to access the RX and TX pins of the HC6812.

J18 (Port P) – J18 contains the 6812's Port P pins (also sometimes refereed to as Port W, for PWM). See schematics for exact pin definitions.

J4 (Port AD) – J4 contains the 6812's analog-to-digital (AD or A/D) port, Port AD. Bits 1:0 must be grounded upon reset for D-Bug12 to run. See schematics for exact pin definitions. This is not yet implemented in D-Bug4744. When it is implemented, D-Bug4744 will use Port AD, bit 7.

J22 (Port T) – J22 contains the 6812's Port T. See schematics for exact pin definitions.

J17 (Latched Address) – J17 contains the 16 bit latched address output from the 74'373's.

J30 (Data) – J30 contains the 8 bit data I/O pins from the 6812. This header is not buffered and is connected directly to the 6812, therefore it also contains the raw upper 8 bit address.

J16 (CPLD I/O) – J16 contains all the CPLD I/O lines for general purpose use including CPLD special function input pins 1 (input/CLR), 2 (input/CLK2), 43 (input/CLK1), and 44 (input/OE1)..

J31 (Port S) – J31 contains the 6812's Port S.

J32 (Port DLC) – J32 contains the 6812's Port DLC.

J25 A/B and J26 A/B (EEPROM & EPROM) – Jumpers are used to select between using EEPROM or EPROM in socket U1. These are implemented as the footprints of surface mount resistors. Connecting the pads with solder is the equivalent of installing a jumper.

4.0 Jumper Definitions

This section provides a detailed description of the jumpers on the board as well as their functions. See Fig. 1. Jumpers are implemented as the footprints of surface mount resistors. To connect or disconnect a signal, simply drag a trace of solder across the pads or break said trace with a hot soldering iron.

J1 (ROM OE) – J1 connects the ROM's output enable signal (OE) to the CPLD via pin 26 on the CPLD. If this jumper is removed, the ROM's OE signal is left floating. **Normally: Installed**

J25 A/B and J26 A/B (EPROM & EEPROM) – These jumpers are used to select between using EEPROM or EPROM in socket U1. For EPROM, J25A and J26A should be connected. For EEPROM, J25B and J26B should be connected. One (and only one) set of these jumpers must be installed. **Normally: One set installed**

J20 (BKGD) **–** This jumper selects the Background Debug Mode. Note: If you install this jumper, you must remove it after reset. This must be done because BKGD is also as a communication line in the BDM header (J14). **Normally: NOT installed**

(Vfp) – Flash Programming Jumper. If installed, connects +12V from Header J10 to VFP pin on 6812. **Normally: NOT installed**

J3 (SA) – This header is used to enable standalone mode on the board. When this jumper is open, PLDC0 is high. When the jumper is connected, PLDC0 is pulled low. The Monitor program can be edited to poll this signal and begin a running pre-described code in memory. [Note that it is mis-labeled as LDC. It should be DLC (Data Link Communications).] **Normally: NOT installed**

UF 68HC12 Development Kit (Board Version 4.2) Manual

J9 (MOD A) – J9 is the Mode A input to the HC12. **Normally: Installed**

J8 (MOD B) – J8 is the Mode B input to the HC12. **Normally: NOT installed**

Power Source	**RP (USB_EN)**
Reg 5V on J11	**NOT** Installed
USB	Installed

Figure 3: Power Source

RP (USB_EN) – This allows the user to select between which power input will be used. If a regulated power source is used via header J11, RP must be disconnected. If you wish to power the board through the USB cable, a solder short should be installed across RP. See Figure 3. **Normally: Installed**

J5 (CPLD Jumpers) – J5 contains a variety of control signals to connect to the CPLD. Each signal is labeled on the board next to the pin it controls. When the jumpers are connected, the signals are connected to the CPLD pins according to Figure 4. All the address signals come from the **latched address** source, thus, they are not the raw address signals. Typically, all the jumpers are installed **except** those on XIRQ and IRQ.

6812 Signal	**CPLD Pin**
DBE	21
A12	20
A13	19
A14	18
A15	17
Reset	16
R/W	14
ECLK	12
XIRQ	11
IRQ	9

Figure 4: J5 Connections

Note: Students should determine which signals to connect for proper memory decoding of the EPROM/EEPROM in the memory map. All decoding circuitry is found in the CPLD.

5.0 Memory Map & Interrupt Vectors

Single Chip Mode

Address (Hex)	**Devices**
0-1FF	6812 Internal Registers
800-BFF	1K Internal SRAM
D00-FFF	768 Bytes of Internal EEPROM
8000-FFFF	*32K Bytes Flash Memory (~100 Writes Only)

Expanded Mode

Address (Hex)	**Devices**
0-1FF	6812 Internal Registers
800-BFF	1K Internal SRAM
D00-FFF	768 Bytes of Internal EEPROM
8000-FFFF	*External 32K space. We will add an 8K EPROM from E000-FFFF.

***Note:** Lower 8K in Single Chip Mode is the same as the lower 8K in Expanded Mode. The upper 32K in Single Chip Mode is Flash Memory and in Expanded Mode this area is open for attaching new devices (which is why we chose to place EPROM here).

Internal SRAM in both Single-Chip and Expanded Modes

Address (Hex)	**Explanation**
800-84A	D-Bug 4744 Monitor’s pseudo-vector locations in Internal SRAM
84B-881	D-Bug 4744 Monitor’s internal SRAM (for version 2.4)
882-8FF	Internal SRAM available to user with D-Bug4744 (often used for stack)
900-BDF	Internal SRAM available to user with D-Bug4744
BE0-BFF	D-Bug4744 stack space

CPLD ROM Decoder Logic Equation:

ROM_OE = ~RESET * RW * DBE * A15 *A14 *A13

The below equation has also been verified to work:

ROM_OE = ~RESET * RW * ECLK * A15 *A14 *A13

Note that OE, RESET, and DBE are active-low and RW = R(H) = W(L) = R/~W . When you build the circuit to create this equation, remember that activation level mismatches create NOTs for free. The ROM will only send outputs to the data pins during read operations when DBE is true and the addresses are appropriate for the ROM (from $E000-$FFFF).

Interrupt Vectors (6812 Hard Designations) and Interrupt Pseudo-vectors

Table 1: 6812 interrupt vectors.

Address (Hex)	Devices
FF80-FFC1	Reserved for future use
FFC2-FFC9	Reserved for mfg. test
FFCA, FFCB	Pulse Accumulator B Overflow
FFCC, FFCD	Modulus Down Counter Underflow
FFCE, FFCF	Reserved for mfg. test
FFD0, FFD1	BDLC
FFD2, FFD3	Analog-to-Digital
FFD4, FFD5	Reserved
FFD6, FFD7	SCI #0
FFD8, FFD9	SPI Serial Transfer Complete
FFDA, FFDB	Pulse Accumulator Input Edge
FFDC, FFDD	Pulse Accumulator Overflow
FFDE, FFDF	Timer Overflow
FFE0, FFE1	Timer Channel #7
FFE2, FFE3	Timer Channel #6
FFE4, FFE5	Timer Channel #5
FFE6, FFE7	Timer Channel #4
FFE8, FFE9	Timer Channel #3
FFEA, FFEB	Timer Channel #2
FFEC, FFED	Timer Channel #1
FFEE, FFEF	Timer Channel #0
FFF0, FFF1	Real Time Interrupt
FFF2, FFF3	IRQ
FFF4, FFF5	XIRQ
FFF6, FFF7	SWI
FFF8, FFF9	Reserved for the Future
FFFA, FFFB	COP Failure Reset
FFFC, FFFD	Clock Monitor Failure Reset
FFFE, FFFF	***Reset**

Table 2: 6812 interrupt pseudo-vectors.

Pseudo-Vector Address	Interrupt Pseudo-Vector Description
$0809-$080B	Analog-to-Digital
$080C-$080E	Serial Communications Interface (SCI)
$080F-$0811	Serial Peripheral Interface (SPI)
$0812-$0814	Pulse Accumulator Input Edge
$0815-$0817	Pulse Accumulator Overflow
$0818-$081A	Timer Overflow
$081B-$081D	Timer Channel 7
$081E-$0820	Timer Channel 6
$0821-$0823	Timer Channel 5
$0824-$0826	Timer Channel 4
$0827-$0829	Timer Channel 3
$082A-$082C	Timer Channel 2
$082D-$082F	Timer Channel 1
$0830-$0832	Timer Channel 0
$0833-$0835	Real Time Interrupt (RTI)
$0836-$0838	IRQ
$0839-$083B	XIRQ
*$FF6C	Software Interrupt (SWI)
$083F-$0841	Unimplemented Opcode Trap
$0842-$0844	COP Failure Reset
$0845-$0847	Clock Monitor Failure Reset
*$FF80	Reset

Note: On the UF 68HC12 Development Kit, the addresses in FFFE & FFFF in external (added on) EPROM/EEPROM point to the external EPROM/EEPROM location of the D-Bug4744 (UF-6812 Monitor/Debugger). Thus, at this location in EPROM/EEPROM there is a vector that is loaded into the PC that causes program flow to begin at the first location in EPROM/EEPROM.

General Hints and Tips

- When the 6812 is reset into any normal mode, the COP watchdog timer is automatically enabled. To disable it write $00 to the COPCTL register at address $0016.
- When the 6812 is reset into any normal mode the R/W disabled externally and the pin is pulled high. To enable the R/W signal write $04 to the PEAR register at address $000A
- The two items above are already performed in the D-Bug4744 (UF-6812 Monitor/Debugger)

6.0 CPLD/Header Cross Reference Table

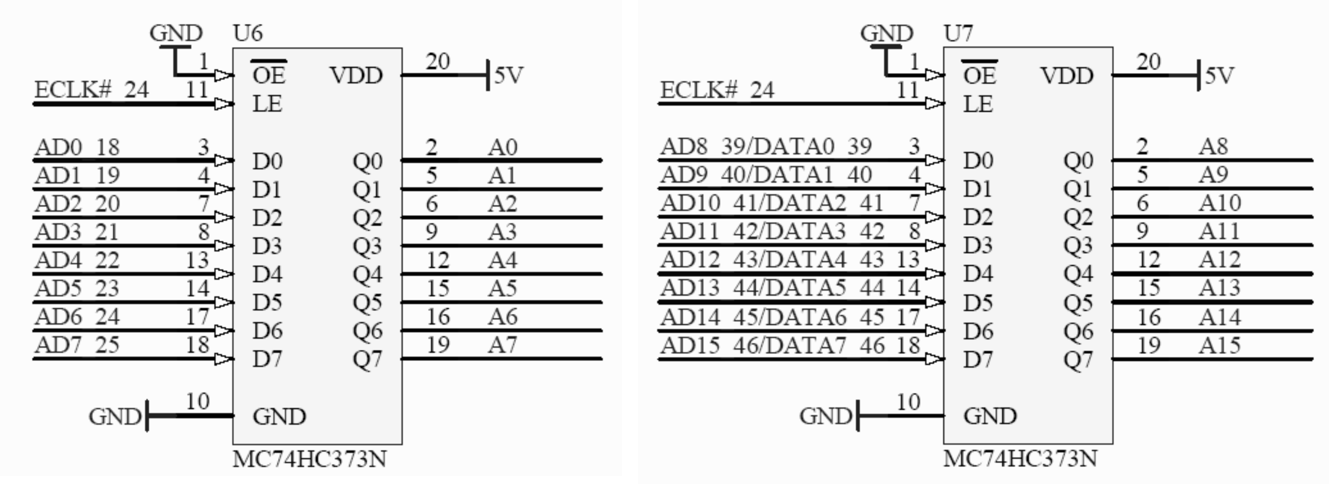
The following table should prove very useful when adding memory-mapped components to the board. It provides a quick reference between the CPLD pin numbers (as seen in Quartus) and the CPLD I/O header (J16) pin numbers that will be will be wire-wrapped to on the board. The hard-wired signals require the proper jumper be installed on the CPLD jumper header (J5).

Table 3: CPLD pin definitions.

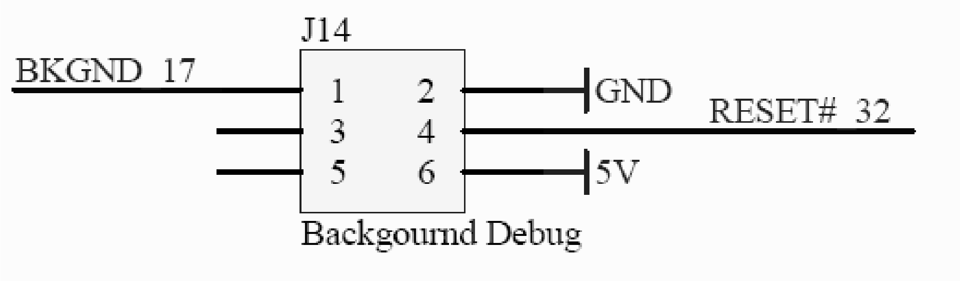
	CPLD Pin #	CPLD I/O Header Pin #	Hard-wired Signal & Type (if any) relative to the CPLD
	4	1	
	5	3	
	6	5	
	8	7	
	9	9	IRQ(L), Output
	11	11	XIRQ(L), Output
	12	13	ECLK, Input
	14	15	R/~W, Input
	16	17	RESET(L), Input
	17	19	A15, Input
	18	21	A14, Input
	19	23	A13, Input
	20	25	A12, Input
	21	27	DBE(L), Input
	24	28	~ECLK=ECLK(L), Output
	25	26	
	26	24	Boot ROM Enable, Output
	27	22	
	28	20	
	29	18	
	31	16	
	33	14	
	34	12	
	36	10	
	37	8	
	39	6	
	40	4	
	41	2	

Note: On the UF 68HC12 Development Kit, the CPLD's ~ECLK (CPLD pin 24, CPLD header pin 28) is already routed to each of the 373's latch enable (LE on pin 11).

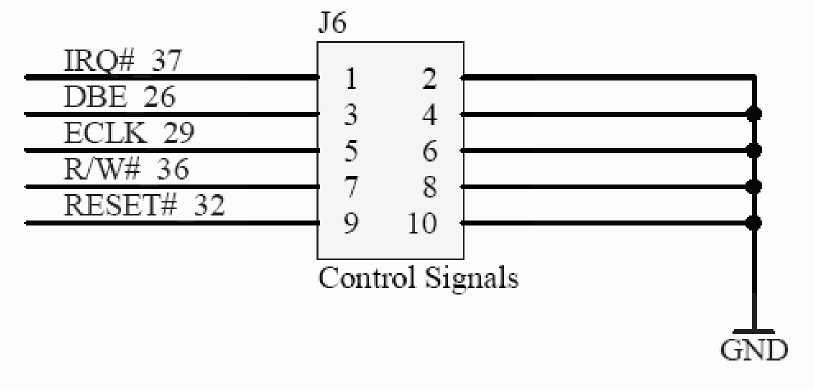
7.0 Schematics



U6 and U7: Address Latches



J14: Background Debug Mode (BDM)



J6: Control Signal Header

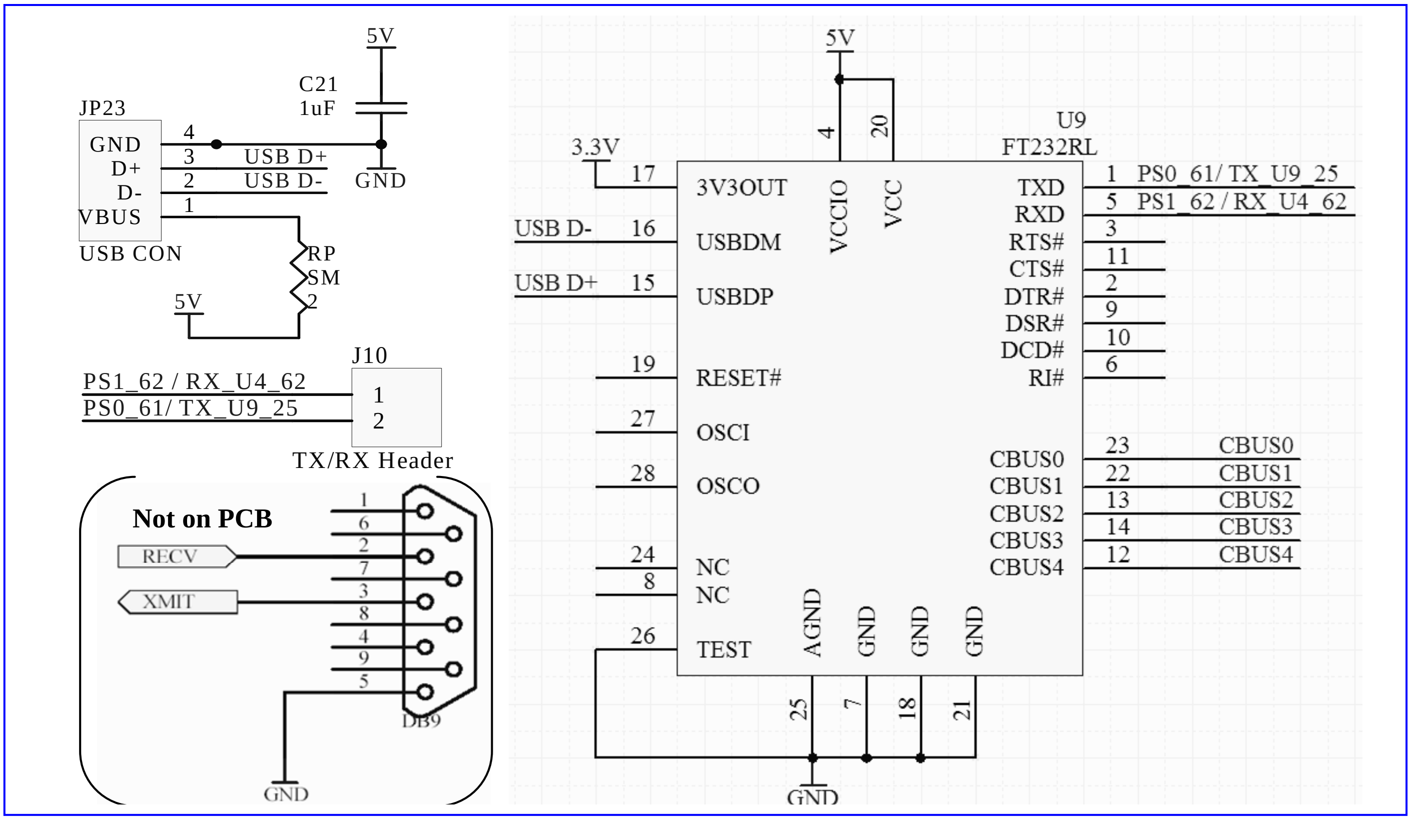
FT232RL Information

The FT232RL is a USB 2.0 to UART Bridge from FTDI (http://www.ftdichip.com/). This chip is capable of taking full-speed data (12Mbps) from a USB cable and converting it into serial format using its internal UART. The UART has an internal clock and can provide baud rates up to 1 Mbit/s. The chip works with most operating systems including Windows®98 SE/2000/XP/Vista, MAC OS-8/9, MAC OS-X, and Linux 2.4 and greater. The data sheet on the web at

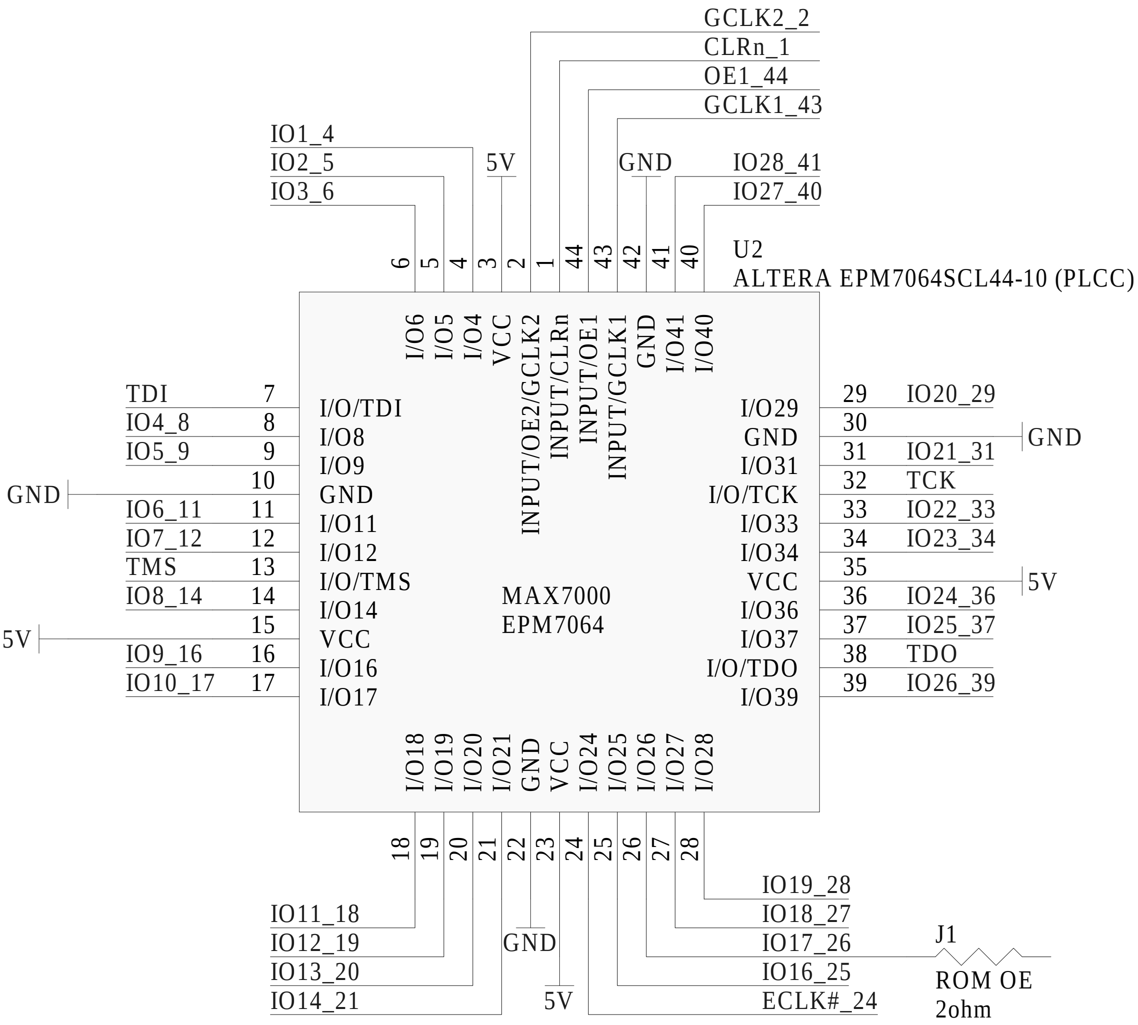
http://www.ftdichip.com/Documents/DataSheets/DS_FT232R.pdf

or or on the class website at

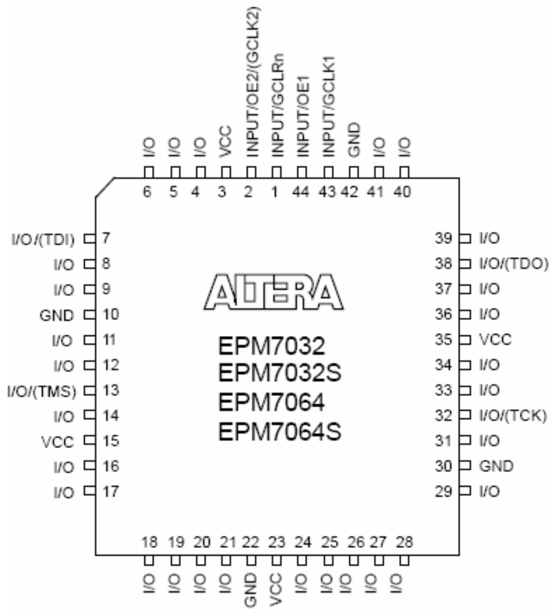
http://mil.ufl.edu/4744/docs/DS_FT232R.pdf :



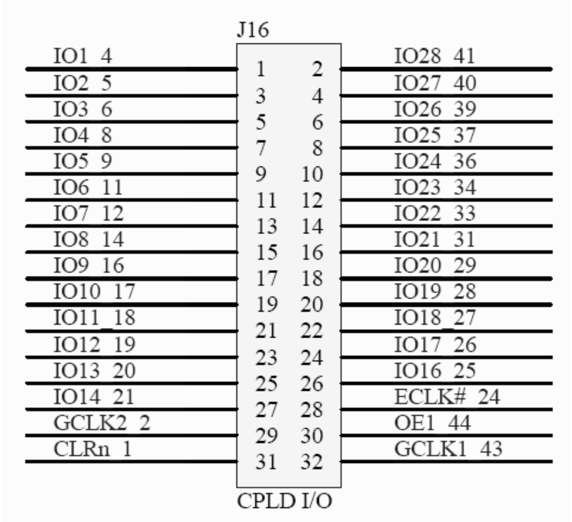
USB/Serial circuits.
1) U9: FTDI USB Bridge (FT232RL); 2) JP23: USB Connector;
3) J10: Serial Port Rx/Tx signals; 4) 9-pin DB9 serial connector (not on PCB)



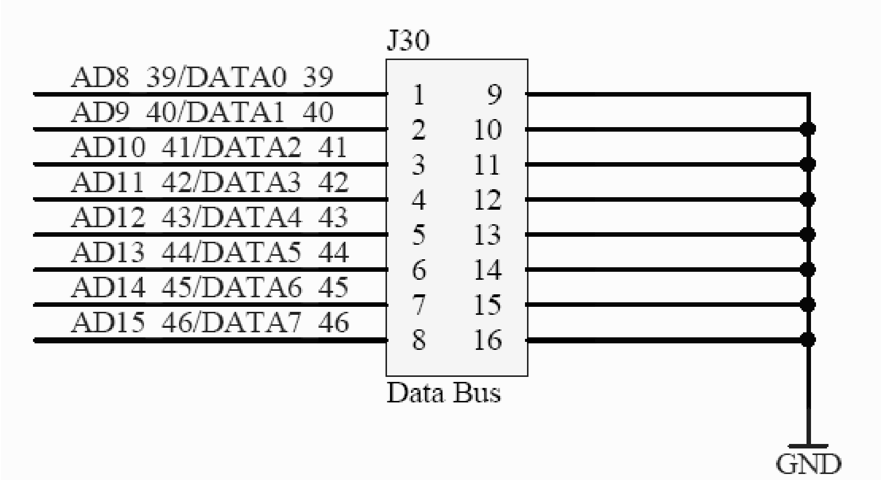
U2: CPLD pin connections. Note that this is **NOT** an accurate pin layout, i.e., the pins on the right side are reversed; i.e., pin 39 should be on the top and pin 29 should be on the bottom.



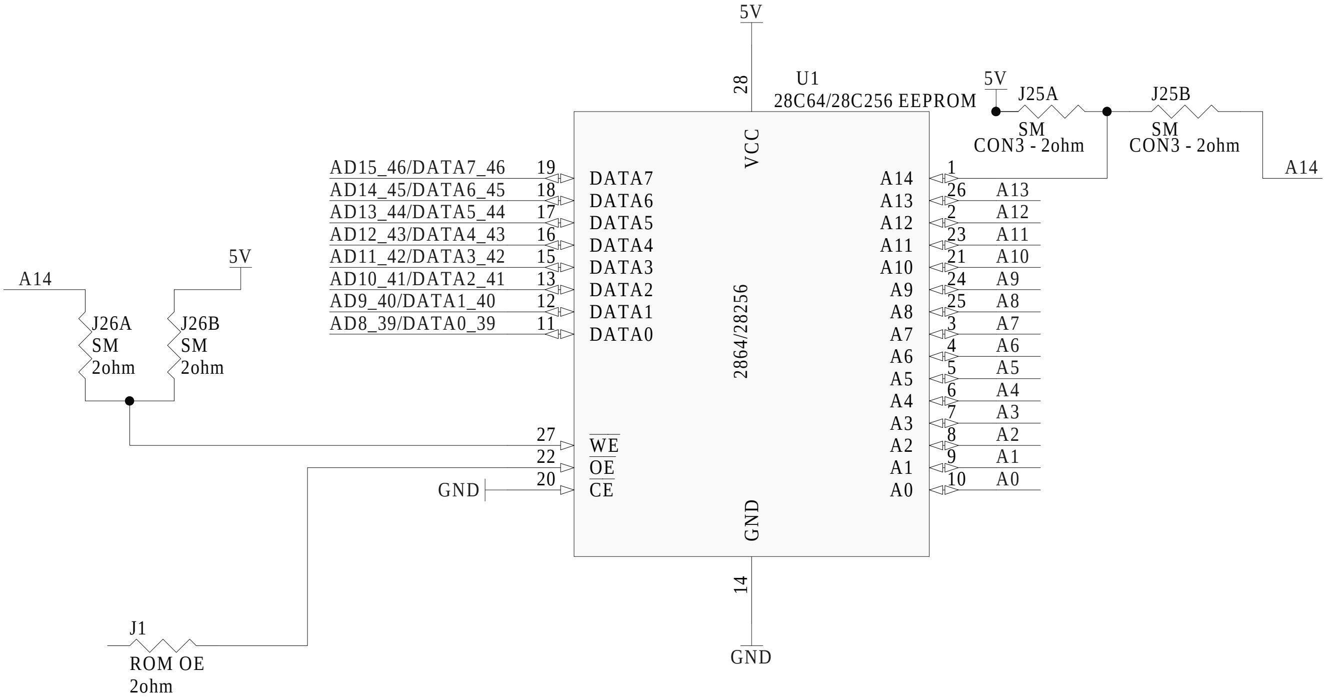
EPM7064 44-Pin Pin-Out Diagram



J16: CPLD IO Header Pin-out. See also U2: CPLD.

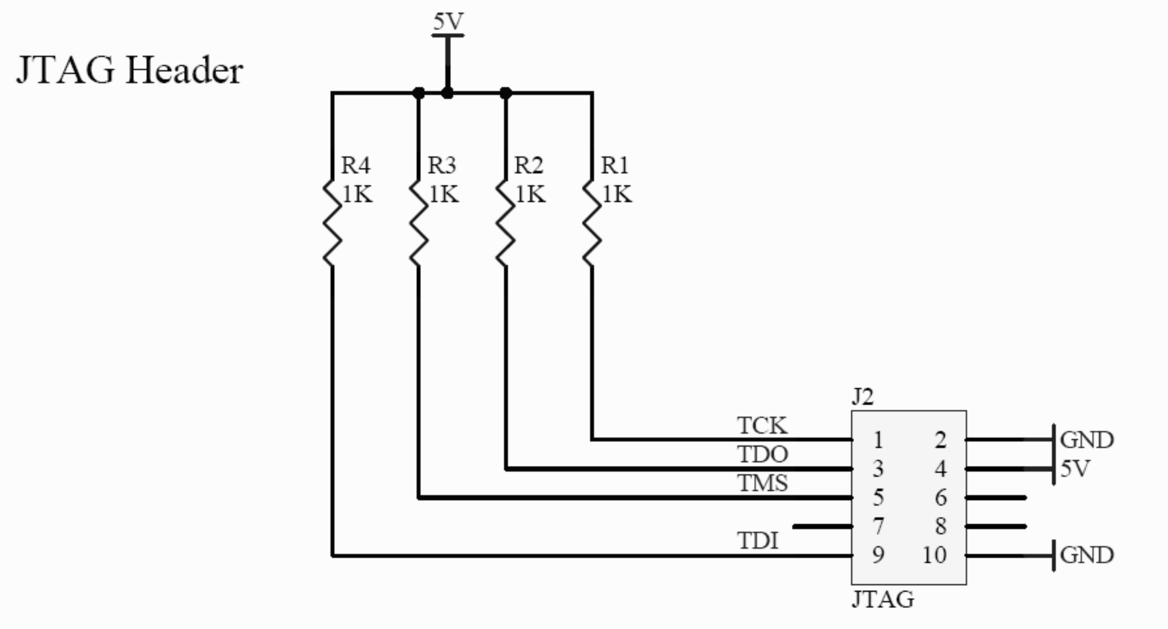


J30: Data Bus

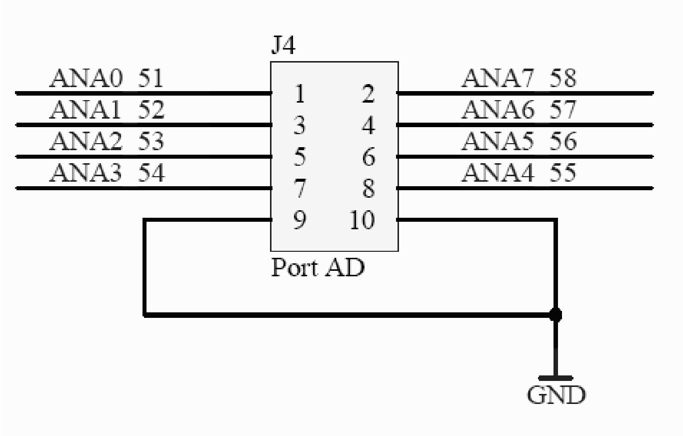


U1: EEPROM (28C64 or 28C256) or EPROM (27C256 or 27C64)

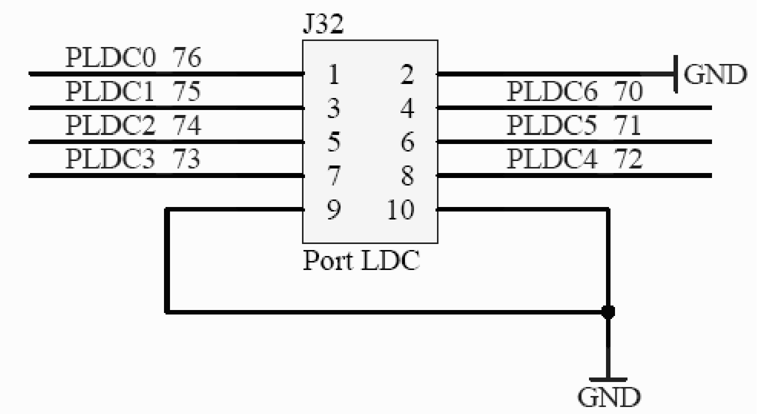
UF 68HC12 Development Kit (Board Version 4.2) Manual



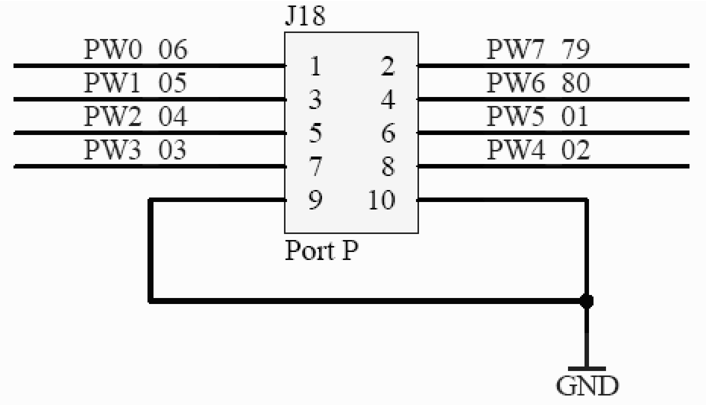
JP2: JTAG Header



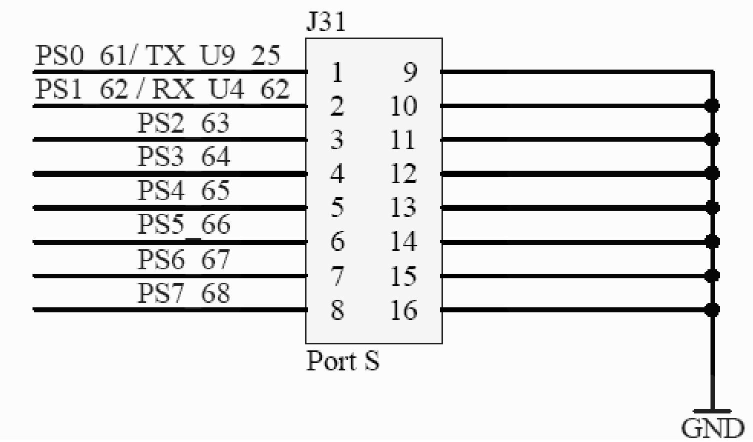
J4: Port AD (Analog-to-Digital) Header



J32: Port DLC (Data Link Communications) Header. Note that it is mis-labeled as LDC.

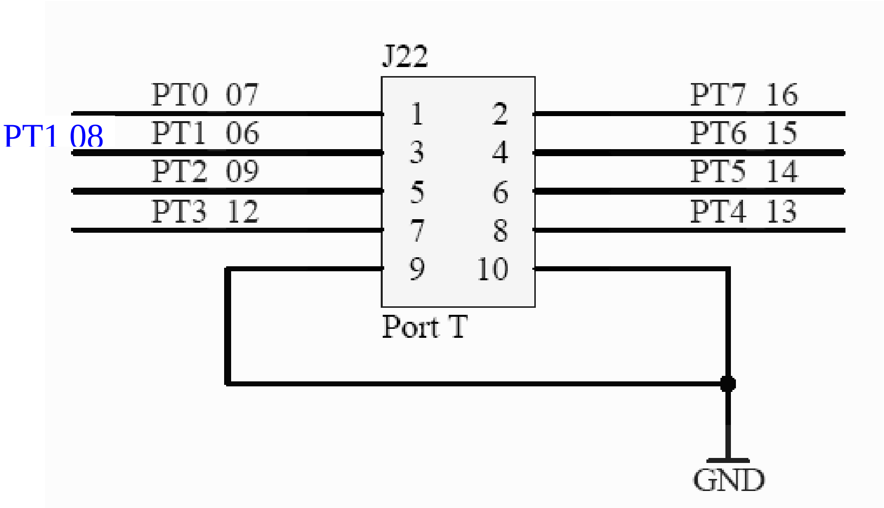


J18: Port P

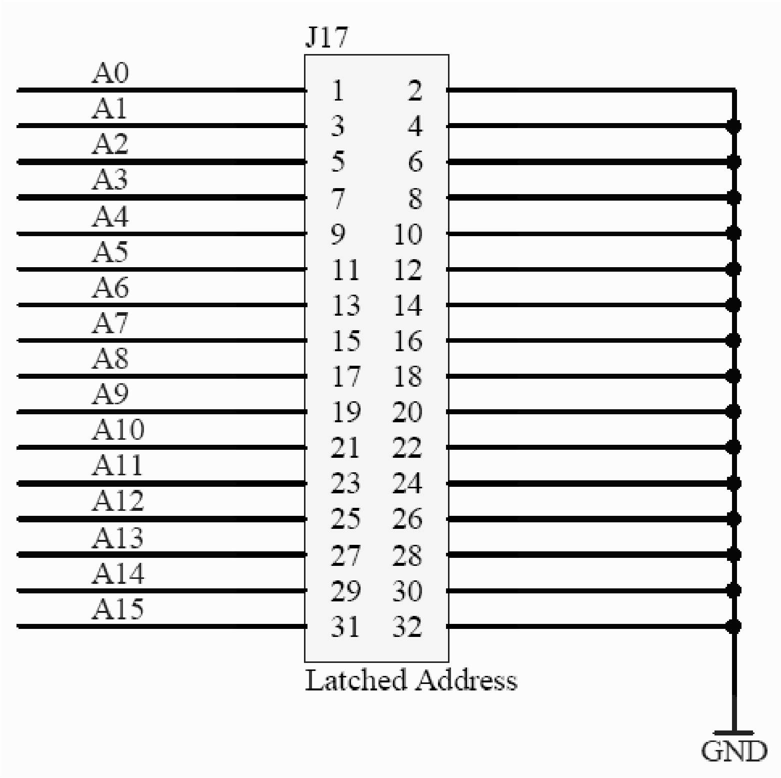


J31: Port S

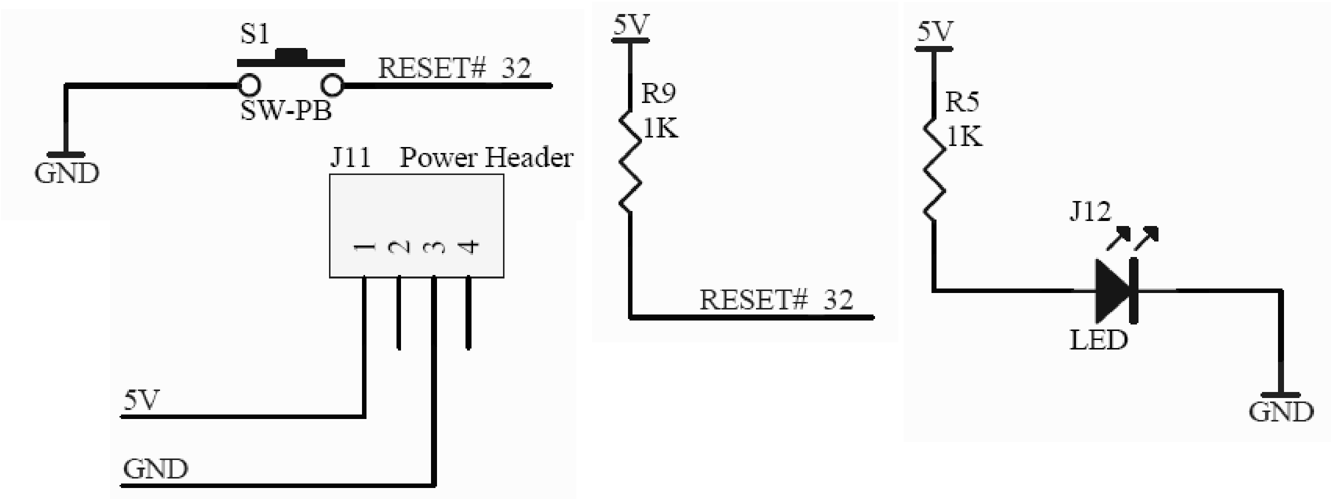
UF 68HC12 Development Kit (Board Version 4.2) Manual



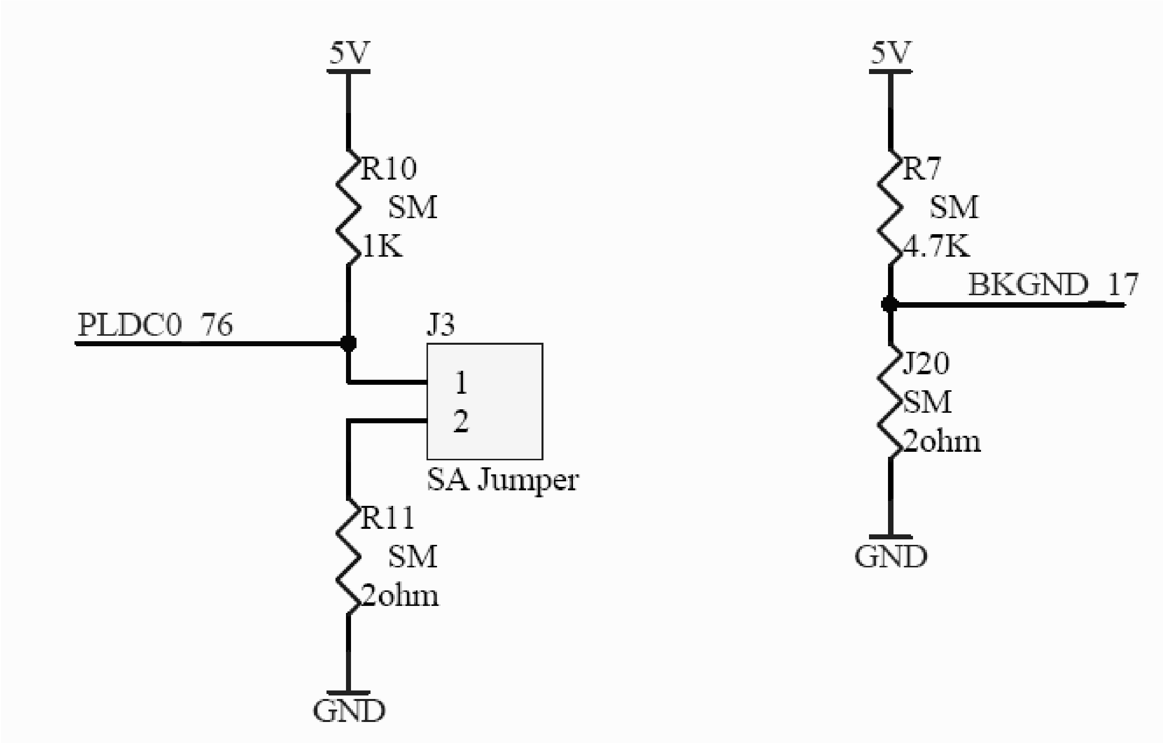
J22: Port T (The net name for PT1 06 should be PT1 08, i.e., pin 8 of the 6812.)



J17: Address Bus

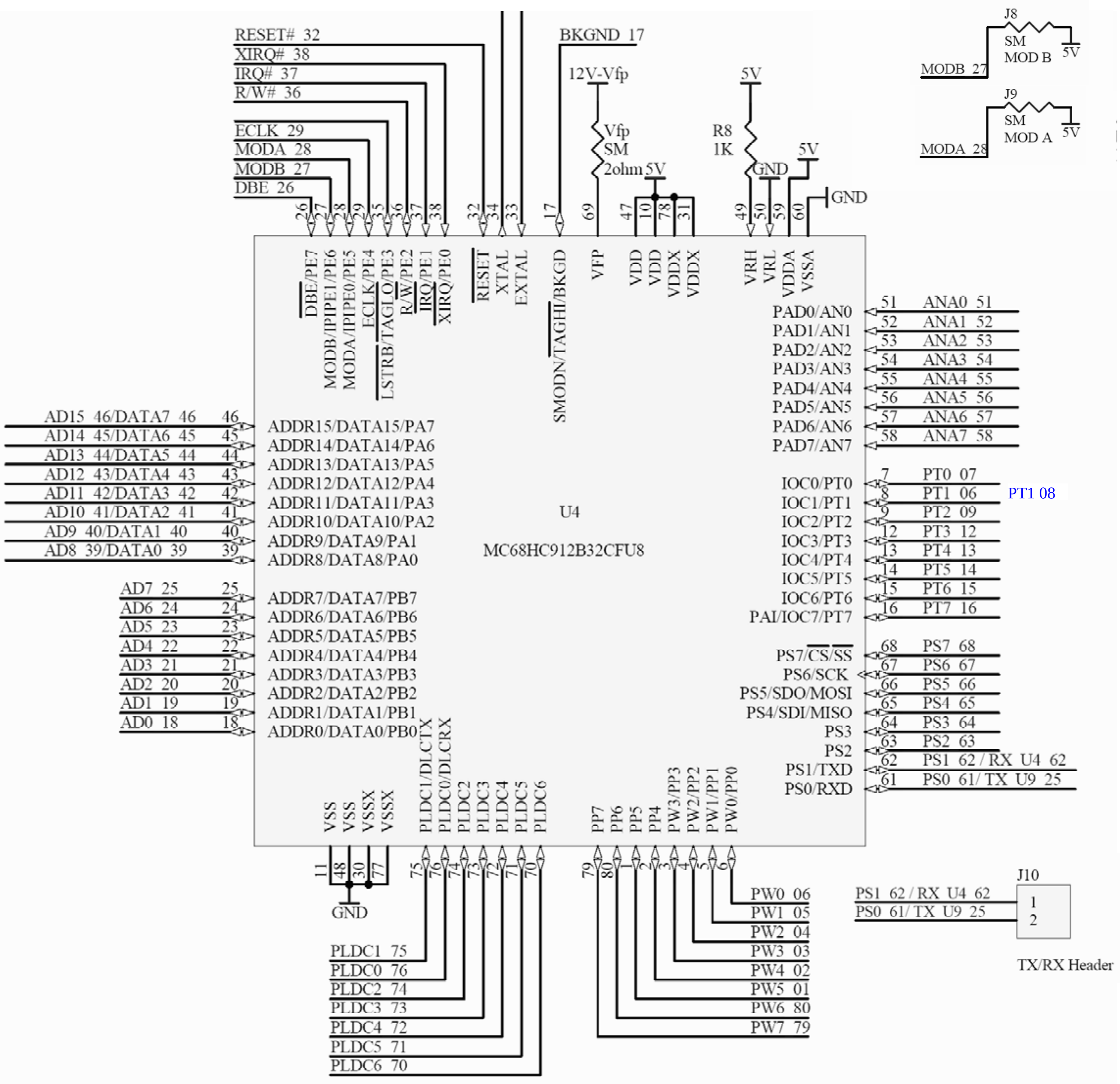


S1: Reset (SW-PB) and Power Header (J11) and Power-On LED (J12) Circuitry



J3 and J20: Stand-Alone Jumper and Background Jumper

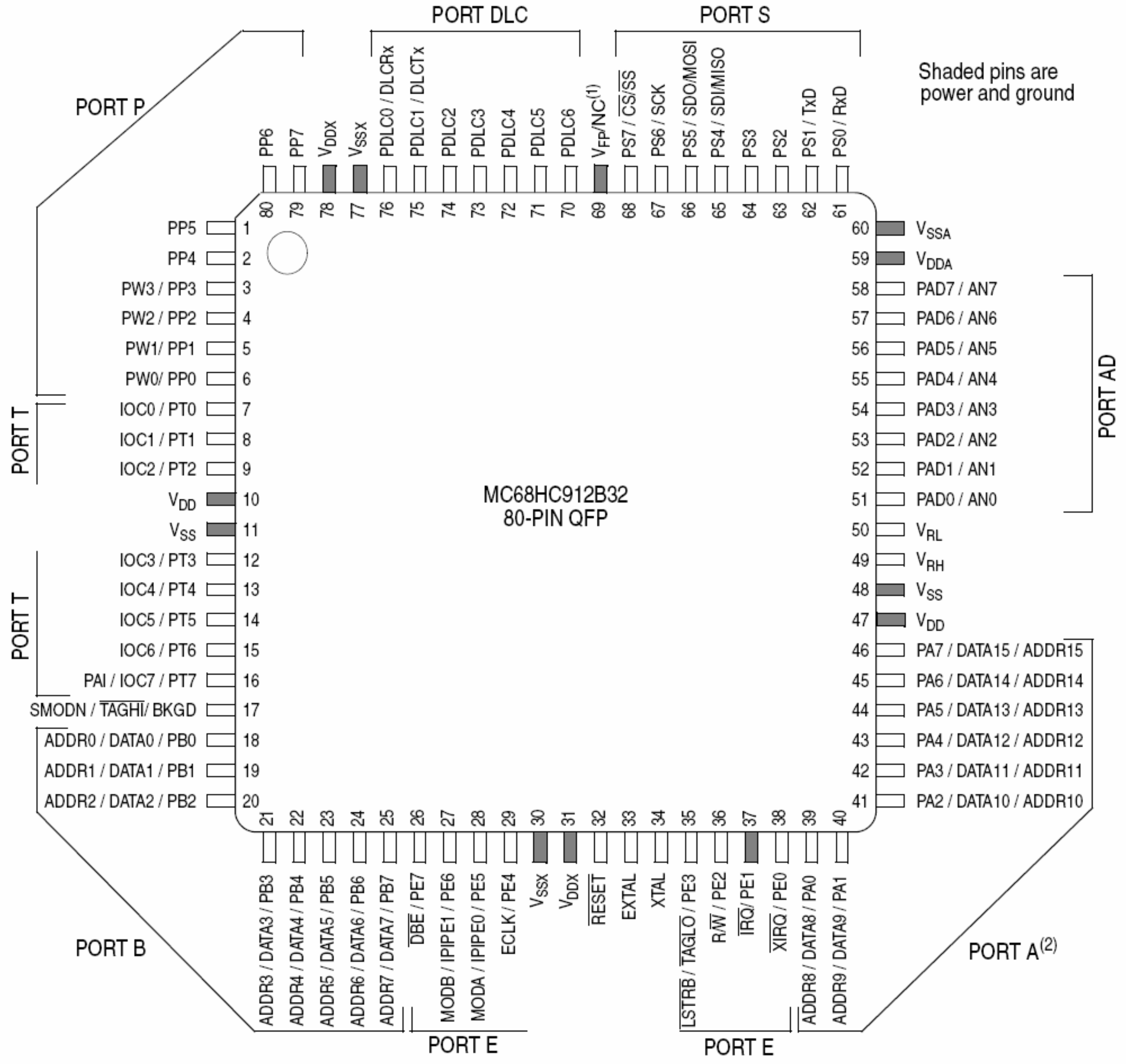
A.1 68HC12B32 Pin-out



U4: 68HC12B32 Pin-out description (**NOT** a physical layout). The layout is shown on the next page.

(The net name for PT1 06 should be PT1 08, i.e., pin 8 of the 6812.)

UF 68HC12 Development Kit (Board Version 4.2) Manual

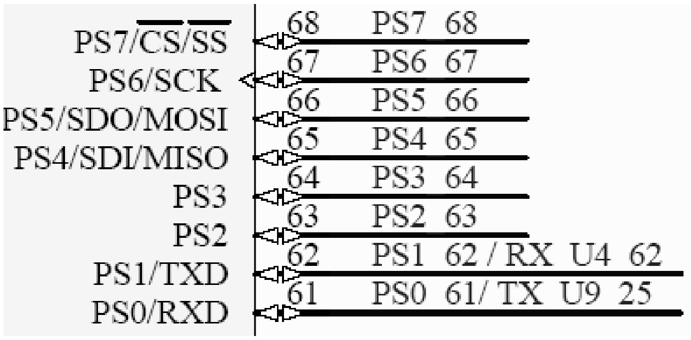


Notes:

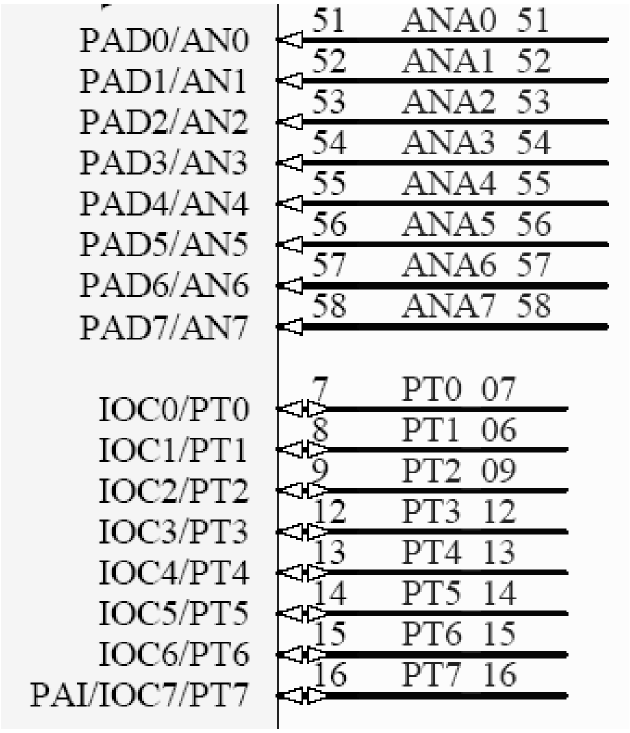
1. Pin 69 is an NC (no connect) on the MC68HC12BE32.
2. In narrow mode, high and low data bytes are multiplexed in alternate bus cycles on port A.

Physical layout and pin assignments of the 68HC12B32 with Freescale names of the signals shown. This is Figure 1-3 taken from the M68HC12B Family Data Sheet (M68HC12B, Revision 9.1).

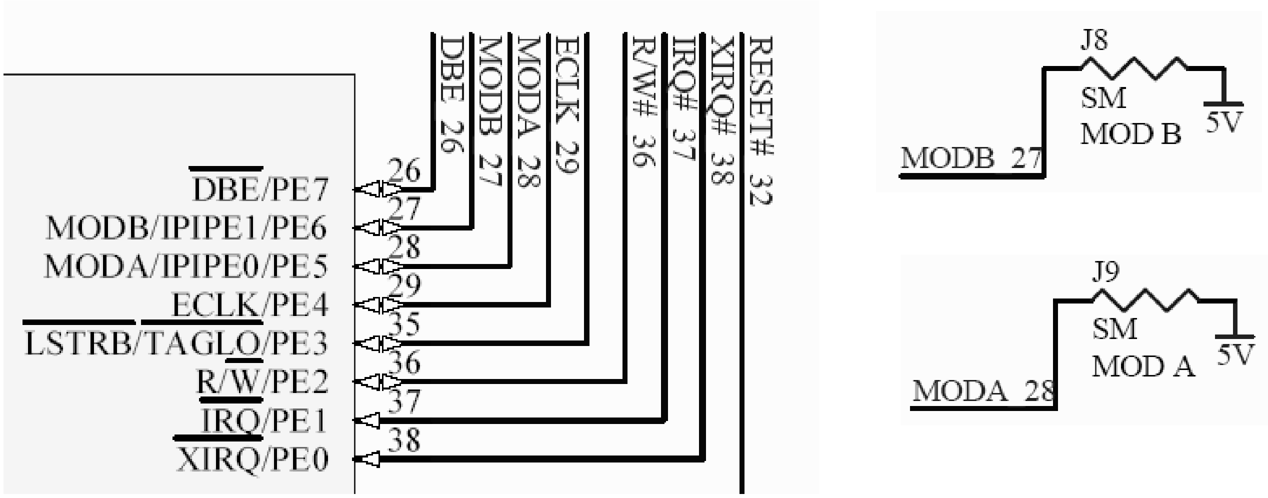
The following figures are magnified sections of the 68HC12 schematic:



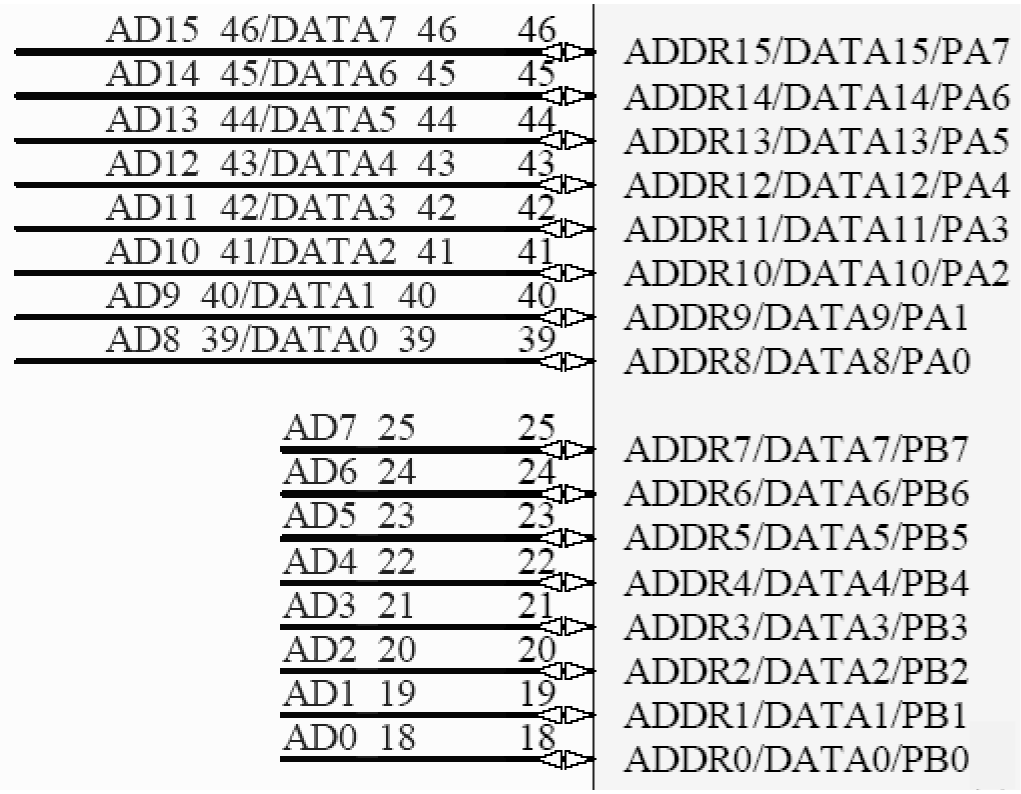
U4: To Port S Header



U4: To Port T (timing) and Port AD (Analog-to-Digital)

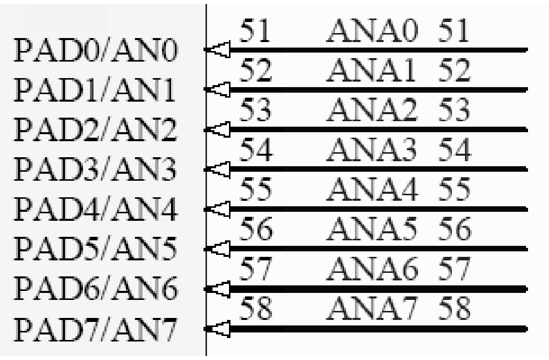


U4: Port E (control) Pins. Note that "#" above means active-low and R/W# means R(H)/W(L).

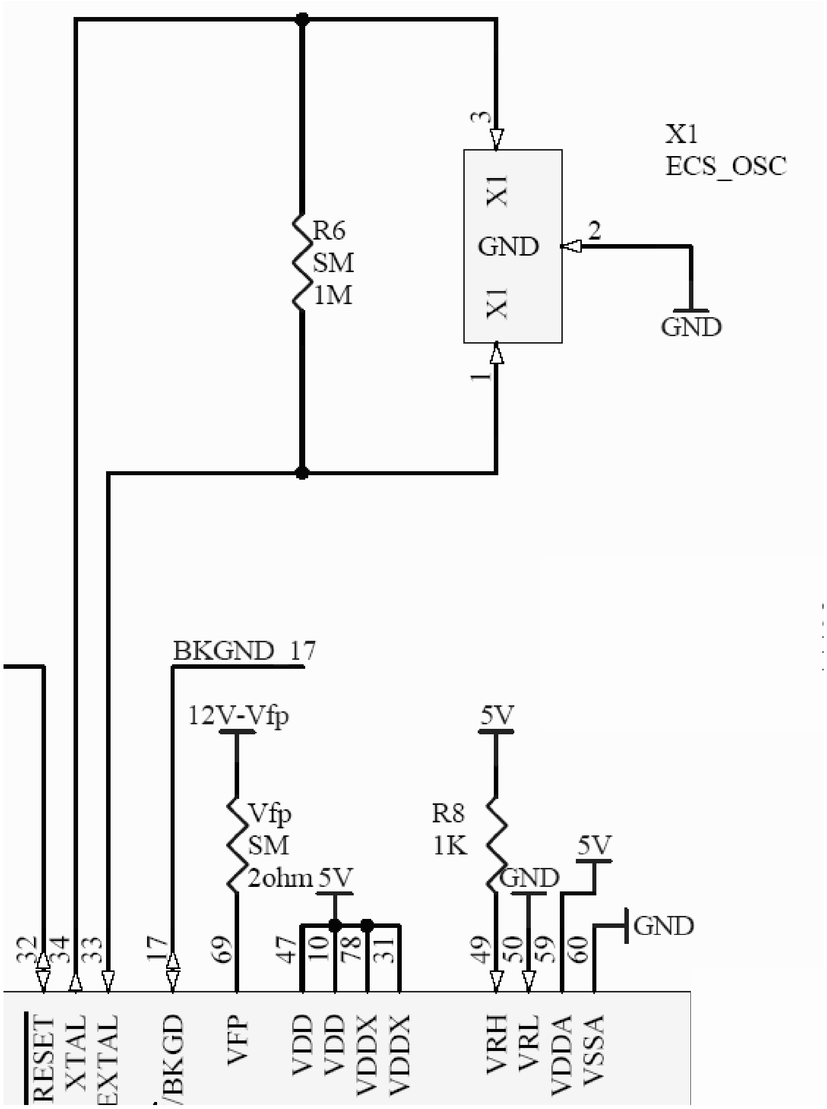


U4: Shared Address and Data Bus

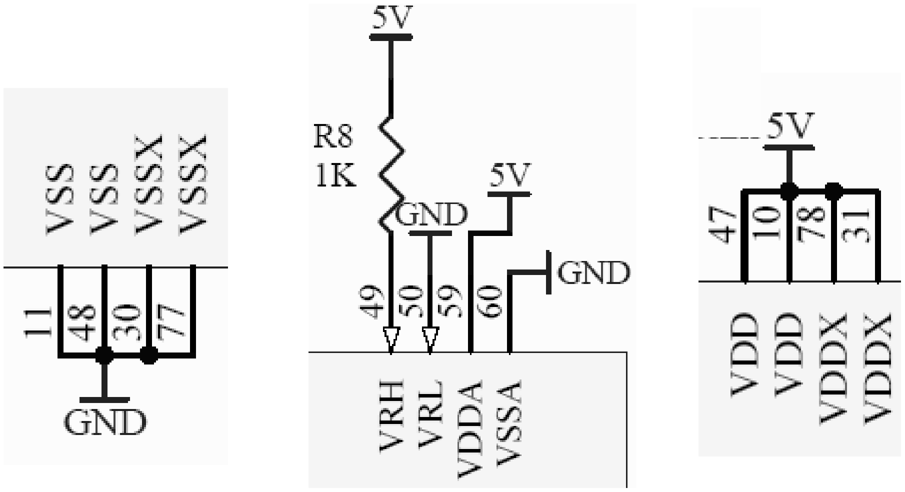
UF 68HC12 Development Kit (Board Version 4.2) Manual



U4: A/D (Analog-to-Digital) Port



U4 and X1: Oscillator



U4: Ground (VSS), Power (VDD), and Analog reference voltage pins

A.2 Altera 7064 – EPROM/EEPROM Decode Equations & CPLD Signal Pin-outs

Required EPROM/EEPROM Equations for 8k device at $E000-$FFFF:

ROM_OE = /RESET * RW * DBE * A15 * A14 *A13
(Some have also used: ROM_OE = /RESET* RW * ECLK * A15 * A14 *A13)
Note that CE, RESET, and DBE are active-low.

ECLK_NOT = /ECLK (ECLK_NOT is also know as ECLK# and ECLK_POUND)

Pin Name & Activation Level	CPLD Pin #
ROM_OE(L)	26
Reset(L)	16
RW [=R/~W=R(H)=W(L)]	14
DBE(L)	21
A15(H)	17
A14(H)	18
A13(H)	19
ECLK	12
ECLK_NOT [ECLK(L)]	24

The above equation and pin-outs enable the EPROM/EEPROM from $E000 to $FFFF in the 6812's memory map. This information can be programmed into the CPLD either as a circuit (using schematic entry into a BDF file) or as VHDL code (with filename extension .vhd) using the Quartus software. A student version of Quartus is available (for free) at http://www.altera.com/.