

Avnet FXT Evaluation Board PPC XMK Design

**Version 1.0
February 2006**

1 Introduction

This document describes a simple PowerPC based design that illustrates the use of Xilinx Micro-Kernel (XMK) operating system.

2 Reference Design Requirements

This reference design will require the following software and hardware setups.

2.1 Software

The software requirements for this reference design are:

- WindowsXP
- Xilinx ISE 10.1 with Service Pack 2
- Xilinx EDK 10.1 with Service Pack 2

2.2 Hardware

The hardware setup for this reference design is:

- Computer with 1 GB RAM and 1 GB virtual memory (recommended)
- Avnet Virtex-5 FXT evaluation board (Rev1)
- Straight through RS232 cable
- Power supply
- JTAG programming cable (USB or PC4)

3 PPC XMK Design Block Diagram

This document describes a simple PowerPC based design that illustrates the use of Xilinx Micro-Kernel (XMK) operating system. Please refer to the “Using Xilkernel” chapter of the Platform Studio User Guide for information on the Xilinx Micro-Kernel operating system. The following figure shows a high-level block diagram of the PPC XMK design.

The design consists of:

- PowerPC processor
- 16KB of BRAM
- 64MB of DDR SDRAM
- RS232 Port
- LED
- Timer
- Interrupt Controller

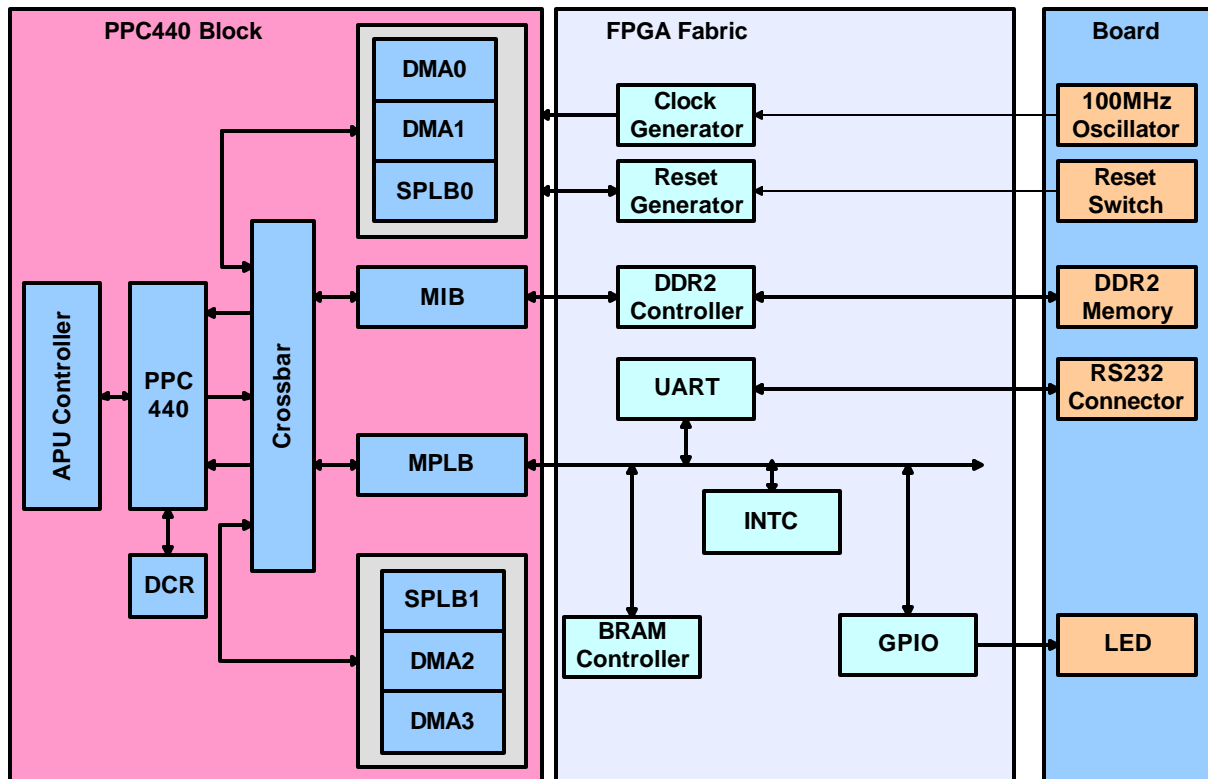


Figure 1 – Reference Design Block Diagram

4 PPC XMK Design Software

The PPC XMK software source code is located in the **/xilkernel_demo** folder of the project directory. The PPC XMK software consists of the following threads:

Thread	Description
<i>shell</i>	This is the main controlling thread and presents a shell with a few simple commands from which you can launch the other demo threads.
<i>prodcon</i>	Producer consumer example thread(s) using message queues.
<i>llist</i>	Linked list demo using the buffer memory allocation interfaces.
<i>sem</i>	Semaphore example showing multiple competing threads using semaphores to coordinate.
<i>TicTacToe</i>	Simple tic-tac-toe game, which illustrates how to dynamically assign stack memory to a thread when creating it.
<i>TimerTest</i>	Simple time management demo.
<i>prio</i>	Thread illustrating dynamically changing priorities and priority queues in the kernel structures.
<i>mutex</i>	Mutex demo, illustrating pthread mutex locks.
<i>clock</i>	Simple thread, using the second timer device and handling interrupts from it, to keep track of wall-clock time. This illustrates user-level interrupt handling.
<i>standby</i>	Simple standby thread.

5 Setting Up the Board

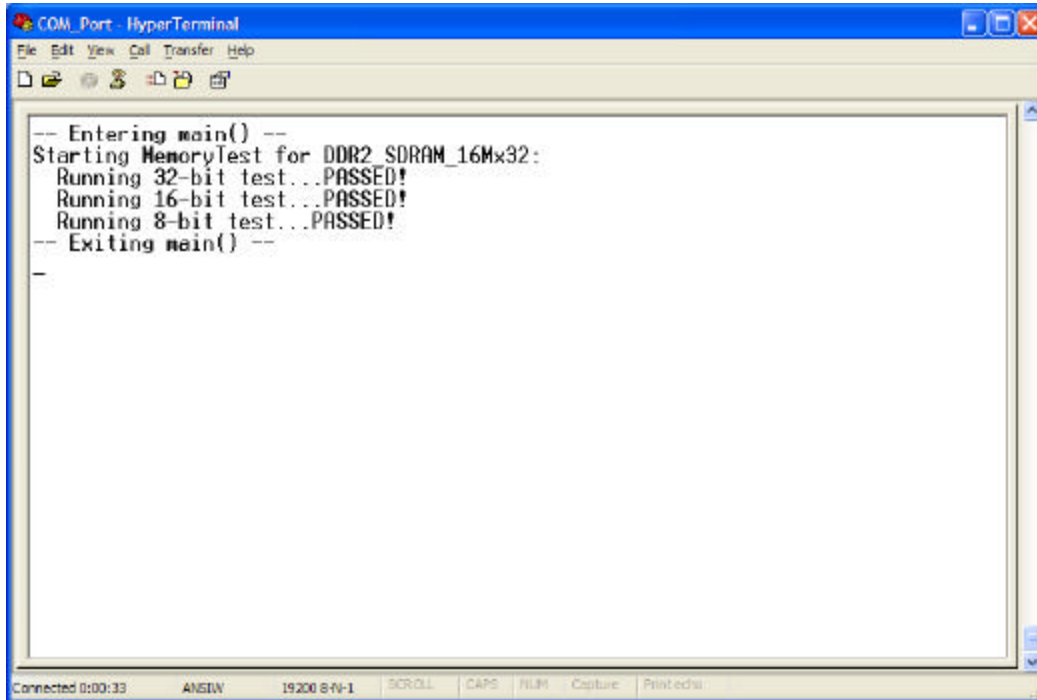
Perform the following steps to setup the board for running the PPC XMK demo.

1. Verify the Power switch, **SW7**, is in the **OFF** position.
2. Install a jumper on JP3 pins 2-3
3. Install a jumper on JP2 pins 2-3
4. Install a jumper on JP1 pins 1-2
5. Install a jumper on JP5 pins 2-3 (FPGA JTAG mode)
6. Connect the power supply to the J11 connector on the FXT evaluation board and also plug it into the AC outlet.
7. Connect the USB JTAG cable to J9 and the USB port of the PC.
8. Connect a straight through RS232 cable to the board DB-9 connector (P1) and the serial port of the PC. Alternatively, you can use an RS232-USB adapter and connect this adapter to the DB-9 connector and the USB port of the PC. In this case, you must install the RS232-USB driver for the adapter.
9. Slide the power switch to the **ON** position

6 Implementing the Design

- Select **Software > Build All User Applications** to compile the software.
- Select **Device Configuration > Update Bitstream** from the XPS GUI to build the design.
- Connect power to the J1 connector.

- Start a Hyper Terminal session and set the serial port parameters to 19200 baud rate, 8 bits, 1 stop bit, no parity and no flow control.
- Select **Device Configuration > Download Bitstream** from the XPS GUI to download the **Memory Test** design to the board. The memory test program will run on the board and you should see the following on the Hyper Terminal.

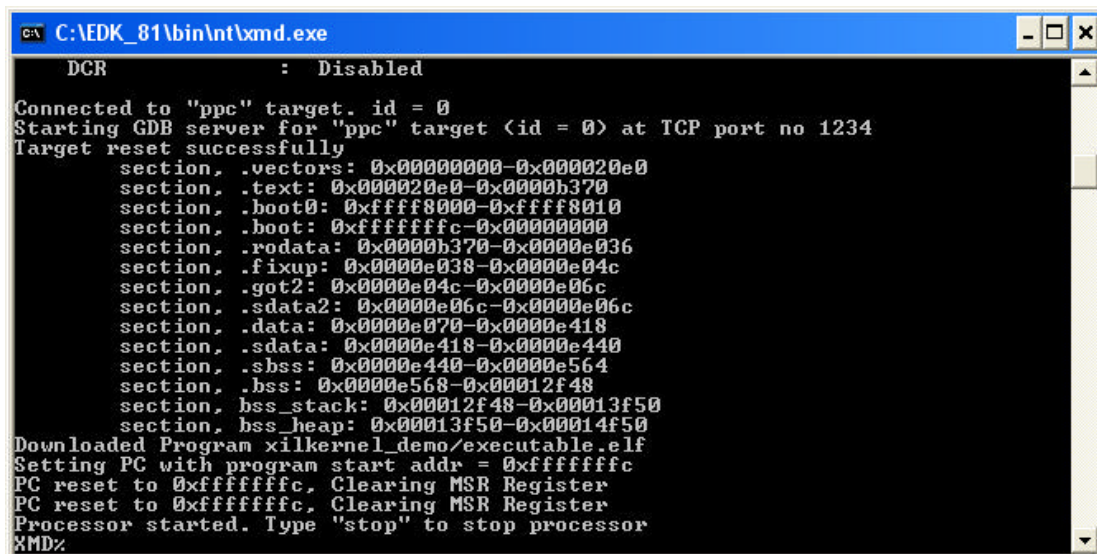


```

-- Entering main() --
Starting MemoryTest for DDR2 SDRAM_16Mx32:
Running 32-bit test...PASSED!
Running 16-bit test...PASSED!
Running 8-bit test...PASSED!
-- Exiting main() --

```

- Select **Debug > Launch XMD** from the XPS GUI to download the PPC XMK software to the external DDR2 SDRAM and run the program. The XMD command window will appear and it should look similar to the window shown in the following figure.

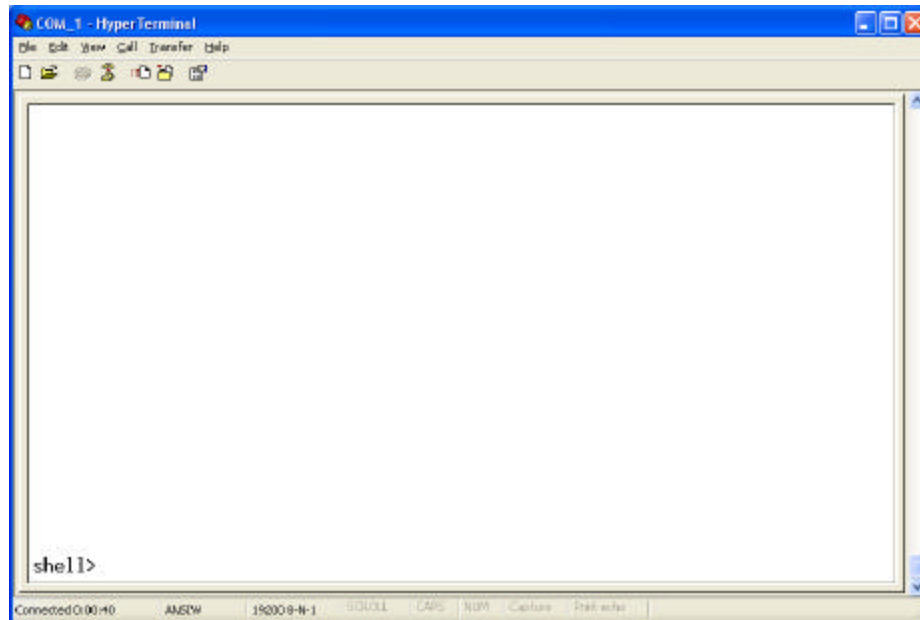


```

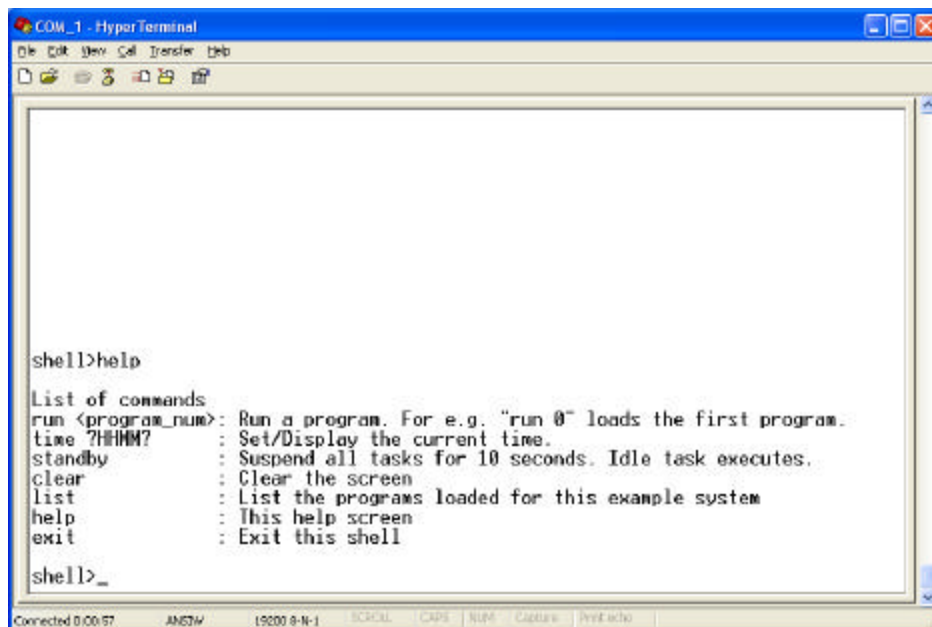
DCR : Disabled
Connected to "ppc" target. id = 0
Starting GDB server for "ppc" target <id = 0> at TCP port no 1234
Target reset successfully
section, .vectors: 0x00000000-0x000020e0
section, .text: 0x000020e0-0x0000b370
section, .boot0: 0xffff8000-0xffff8010
section, .boot: 0xffffffffc-0x00000000
section, .rodata: 0x0000b370-0x0000e036
section, .fixup: 0x0000e038-0x0000e04c
section, .got2: 0x0000e04c-0x0000e06c
section, .sdata2: 0x0000e06c-0x0000e06c
section, .data: 0x0000e070-0x0000e418
section, .sdata: 0x0000e418-0x0000e440
section, .sbss: 0x0000e440-0x0000e564
section, .bss: 0x0000e568-0x00012f48
section, bss_stack: 0x00012f48-0x00013f50
section, bss_heap: 0x00013f50-0x00014f50
Downloaded Program xilkernel_demo/executable.elf
Setting PC with program start addr = 0xffffffffc
PC reset to 0xffffffffc, Clearing MSR Register
PC reset to 0xffffffffc, Clearing MSR Register
Processor started. Type "stop" to stop processor
XMD%

```

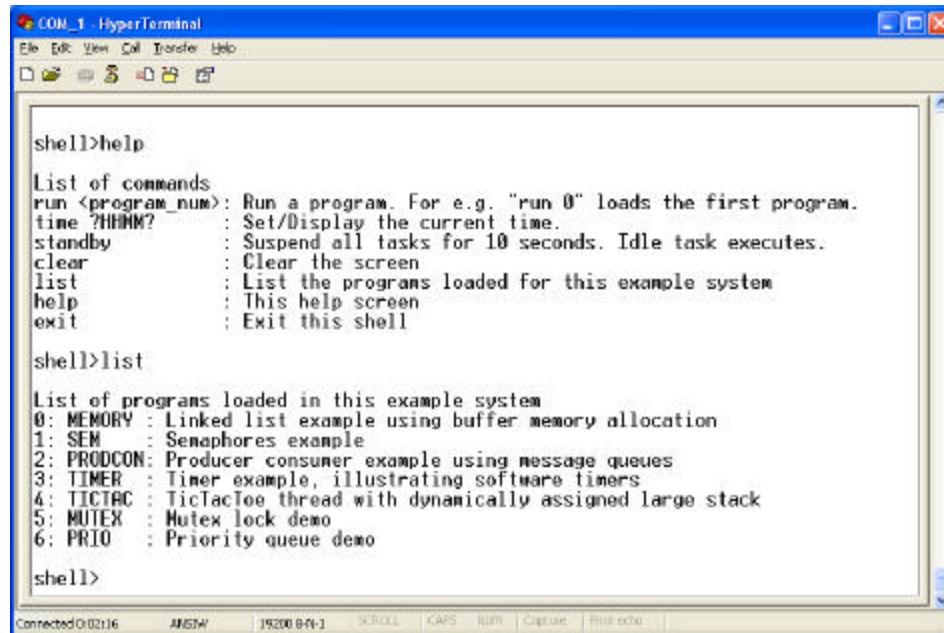
- After successful PPC XMK code download to the board (as shown in the XMD command window), the PPC XMK program will run and you should see the following on the Hyper Terminal.



- Enter **“help”** to get a list of commands.



- Enter **“list”** to get a list of programs.



```

COM_1 - HyperTerminal
File Edit View Call Transfer Help

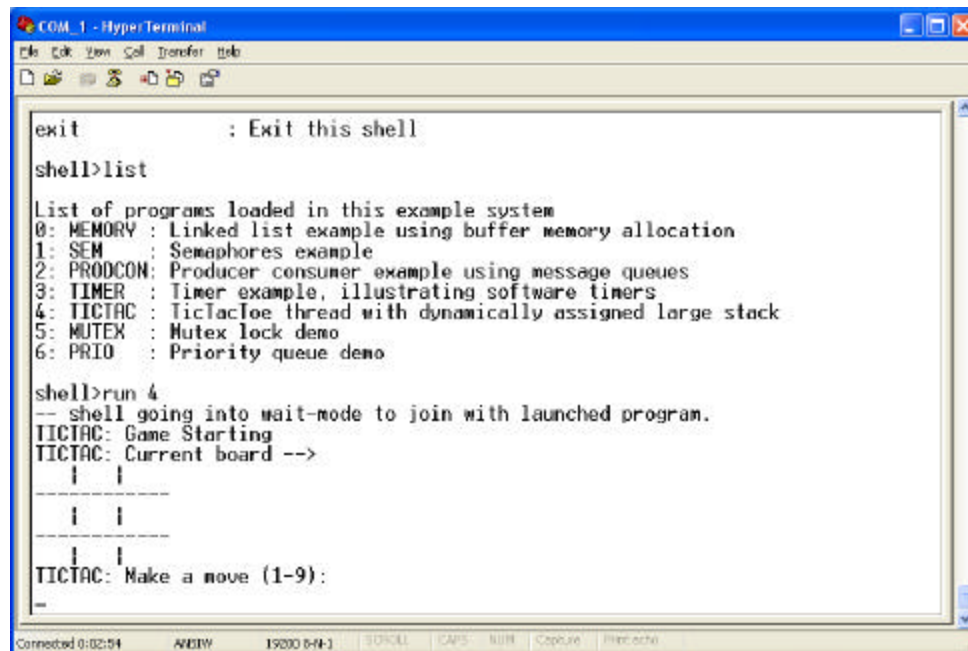
shell>help
List of commands
run <program_num>: Run a program. For e.g. "run 0" loads the first program.
time ?HHMM?      : Set/Display the current time.
standby          : Suspend all tasks for 10 seconds. Idle task executes.
clear            : Clear the screen
list             : List the programs loaded for this example system
help             : This help screen
exit             : Exit this shell

shell>list
List of programs loaded in this example system
0: MEMORY : Linked list example using buffer memory allocation
1: SEM    : Semaphores example
2: PRODCON: Producer consumer example using message queues
3: TIMER  : Timer example, illustrating software timers
4: TICTAC : TicTacToe thread with dynamically assigned large stack
5: MUTEX  : Mutex lock demo
6: PRIO   : Priority queue demo

shell>

```

- Enter **“run”** followed by a number (0-6) to run a program. For example, enter **“run 4”** to play the Tic-Tac-Toe game.



```

COM_1 - HyperTerminal
File Edit View Call Transfer Help

exit          : Exit this shell

shell>list
List of programs loaded in this example system
0: MEMORY : Linked list example using buffer memory allocation
1: SEM    : Semaphores example
2: PRODCON: Producer consumer example using message queues
3: TIMER  : Timer example, illustrating software timers
4: TICTAC : TicTacToe thread with dynamically assigned large stack
5: MUTEX  : Mutex lock demo
6: PRIO   : Priority queue demo

shell>run 4
-- shell going into wait-mode to join with launched program.
TICTAC: Game Starting
TICTAC: Current board -->
  | |
--| |
  | |
  | |
TICTAC: Make a move (1-9):

```