PowerPC™

EXCIMER USER'S MANUAL
Minimal PowerPC 603e Evaluation Board

Motorola RISC Applications
risc10@email.sps.mot.com

This document describes operation of the Excimer PowerPC 603e™ evaluation board manufactured by Motorola. Excimer is an implementation of a minimal PowerPC design as described in the application note AN1769D, "Designing a Minimal PowerPC™ System".

This User's Manual is a very dynamic document which will be updated frequently with Motorola enhancements and customer feedback. Updates are available on our website: http://www.mot.com/SPS/PowerPC/teksupport/teklibrary/index.html

The instructions in this manual are specific to a particular revision of the components in the Excimer kit. The revisions in effect at this writing are shown in Table 1.

MOTOROLA

Table 1. Excimer Kit Component Revisions Addressed in this Document

Component	Revision	Date (if applicable)
Excimer Board (part no PPCEVAL-EXCM3)	X3	
DINK32 User's Guide	Ver 9.0 Rev 5.0	Sep 1998
DINK32 Software	Ver 10.0 Rev 5.0	Dec 1998
Macraigor OCD Interface		

This document contains the following topics:

Part 1 Introduction

The Excimer board is meant to be a low cost evaluation board to show a minimal PowerPC system design. Possible uses for the EXCIMER include:

- PowerPC architecture learning tool
- PowerPC assembly language learning tool
- Embedded C programming learning tool
- University laboratory tool
- Running small benchmarks to investigate performance
- Proof of concept for system design-in

1.1 Features

The EXCIMER board has the following features:

- 1 MByte of RAM (only 512KB on rev X2)
- 4 Mbytes of FLASHROM
- 2 serial ports
- PowerPC 603e microprocessor

- Berg connector I/O interface
- COP connector

The EXCIMER block diagram is shown in Figure 1.

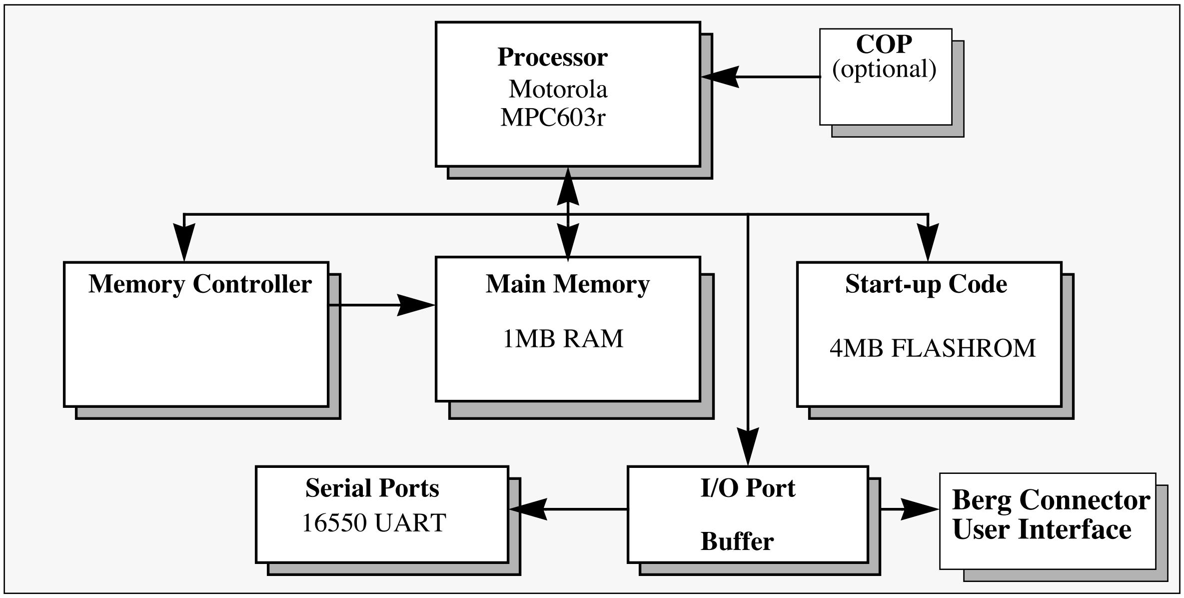


Figure 1. Excimer Minimal System Board

1.2 Excimer Kit

In addition to the EXCIMER board the EXCIMER kit contains the following:

- One power supply
- One serial null modem cable
- One parallel cable
- One Macraigor Systems Inc. Wiggler™
- Several demonstration copies of C compilers/debuggers on CDROM
- PowerPC documentation CD
- *EXCIMER User's Manual* (this document)
- *DINK User's Manual*

1.3 Physical Layout

Figure 2 shows the physical layout of the Excimer board with several important features labeled.

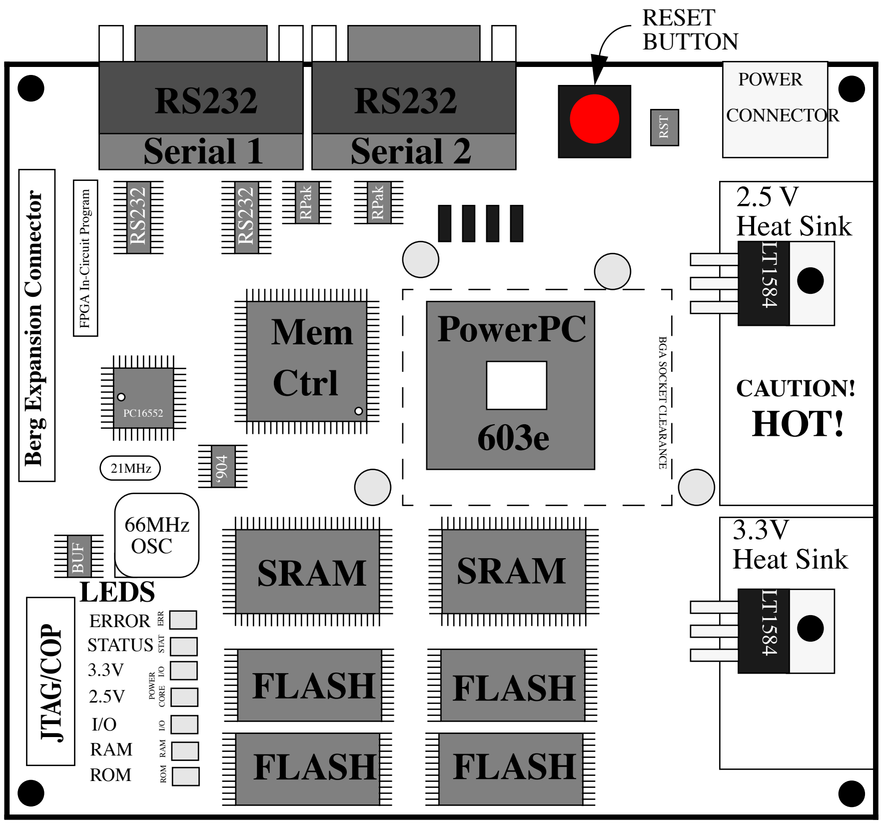


Figure 2. Excimer Minimal System Board

Part 2 Getting Started

To begin using the Excimer evaluation board,

1. Connect one end of the serial null modem cable to the board's serial port 1 and the other end to COM1 on the PC.
2. Launch a terminal emulator program such as Smartcom or HyperTerminal on the PC. Configure for:
   - — 9600 baud
   - — 8 data bits
   - — no parity
   - — 1 stop bit
   - — hardware flow control (RTS/CTS)
   - — VT100 emulation
3. Connect the parallel cable between the parallel port on the PC and the Wiggler interface board.
4. Connect the twenty pin ribbon cable on the Wiggler to the JTAG/COP header on Excimer.

CAUTION: Get the orientation correct! Pin 1 on the cable has a red stripe; Pin 1 on the JTAG connector is labeled. While the connector is designed to be keyed, the cables rarely are.

NOTE: The JTAG connector's pin 1 will be rotated 180 degrees from revision X2 to revision X3 to better accommodate the cable routing.

5. Choose the appropriate international AC power connection (if provided) and plug the power supply into the AC power source.
6. Connect the power supply to the power connector on the board.
7. The EXCIMER will run a POST (power on self test) which will turn off the status and error LEDs. A hardware failure will cause one or both LEDs to stay on. If this occurs refer to Appendix A, “Troubleshooting.”
8. After successfully completing POST, Excimer will begin executing code from the reset vector at 0xFFF00100. The code executed will copy the contents of ROM into RAM and begin executing MDINK, the minimal version of Motorola’s Diagnostic Nano-Kernel (DINK). The user should see an MDINK banner and prompt on the terminal emulator screen.

   MDINK_603e >>

9. The user now has a short interval to halt the boot process and remain in MDINK by hitting any key. From MDINK, you can download upgrades to DINK available from Motorola’s website or other software in s-record format.
10. If the user does not intervene in the boot process by hitting a key, Excimer will proceed to branch to DINK32 and present another banner and the DINK32 prompt on the terminal emulator screen.

    DINK32_603e >>

11. The DINK32 monitor is now available to accept commands. Type in ‘help’. This will show you a list of DINK commands. Help will further explain a command if you type help and a command name. For example ‘help mm’ will explain the use of the memory modify command. For more information on DINK consult the DINK User’s Manual.

Part 3 Programming

To write a C program for Excimer you will need an editor, a compiler, and a linker that will generate Motorola format s-records. Several vendors have offered to provide limited capability, demonstration versions of their compilers and linkers to be distributed with Excimer. Follow the manufacturer’s directions for installing the demo software on your PC.

The following example assumes you have installed Metaware’s High C/C++ PowerPC Embedded Development Toolset. The Limited Evaluation Version 3.1 of this program will operate indefinitely (no limit to the demonstration period) but allows only 256 variable declarations and 256 function declarations. Also, it will only link up to 12 files and the resulting executable must be less than 156KB. These limitations prevent it from building DINK or any large projects but still allow it to be very useful for a minimal evaluation board like Excimer.

1.1 An Example Program

To compile and download a simple program, type the following program into any text editor, e.g. NotePad or WordPad on the PC:

```
/*
 example routine to turn on STATUS LED
*/
void main(void)
```

```
{
        *(char *) (0x40200000) = 0x04;  /* turn status led on */
}
```

Save the file as ledon.c as a <u>text</u> file. (Many text editors save in their own format with extra formatting characters that will cause the compile to fail.) Also, text editors like WordPad in Windows will append a .txt to the filename which must be removed before passing to the compiler.

Assuming you have installed a Metaware compiler on your PC type the following in a DOS window.

```
hcppc -Hppc603 -c ledon.c
```

To load the program and get a downloadable s-record file type the following in the DOS window.

```
ldppc -B start_addr=0x70000 -xm ledon.o
```

A file named ‘a.hex’ file will be generated and this file will be the one to download to Excimer.

In the Excimer terminal window, at the DINK32 prompt type the following:

```
dl -k <return>
```

This is the DINK download command that takes the s-record file and loads it into RAM on the EXCIMER board. Downloading to DINK is equivalent to typing entries into the keyboard. In fact, text files of keystrokes or DINK commands can be downloaded as easily as s-records. The terminal emulation program will provide an autotype function that permits the user to select the a.hex s-record file previously generated and download it through the serial into Excimer.

The DINK32 prompt returns when the download is complete.

Now we can type the ‘go’ command to run the program to turn the led on.

```
go 70000
```

The Status LED on Excimer is controlled by the $\overline{\text{OUT 2}}$ signal of the Serial 1 UART (National Semiconductor PC1655D Dual Universal Asynchronous Reciever/Transmitter). It will turn on when this program writes a one to bit 3 (bit 0 is the LSB) of the MODEM Control Register at address 0x40200000 of Excimer. It will remain on until disabled by another program you write, compile, link and download or until the Excimer board is reset.

If the LED does not go out, the debug capabilities of Excimer and the Wiggler can be used to debug your program.

Part 2 Debugging

Debug on Excimer is supported with the following capabilities in DINK:

- Assembly and disassembly of PowerPC instructions
- Display and modification of registers
- Display, modification and movement of system memory
- Singlestep and continued execution from a specific address
- Display, setting, removal of instruction breakpoints

DINK is an example of a ROM-resident debugger. It resides in ROM and is copied down to RAM for execution. ROM resident debuggers are often criticized because:

- They take up space in the target system memory
- They have to change (save and restore) the context of the user's application to run
- They are ineffective at debugging boot code that initializes the hardware or ROM resident debugger
- They often cannot set breakpoints in ROM

PowerPC microprocessors provide on-chip debugging logic to overcome some of these deficiencies and provide other capability without storing code in memory or having to save and restore the state of the User's program. Specifically, the PowerPC on-chip debugging support includes:

- Run control - the ability to halt, single step, resume execution , or go to a specific address and begin execution.
- Display and modification of registers and processor resources
- Display, modify and download to memory (possibly including on chip memory like cache and/or TLBs)
- Display, setting, removal of instruction breakpoints in hardware

This capability is accessed by sending commands to the common on-chip processor (COP) through the JTAG (Joint Test Advision Group or IEEE1149.1 standard) interface. A connector is provided to access this on Excimer and numerous third parties provide software and hardware to take advantage of it while debugging code. The Excimer kit includes an inexpensive Wiggler JTAG/COP interface from Macraigor Systems, Inc.

1.1 The Wiggler

The Wiggler from Macraigor Systems is an inexpensive JTAG/COP interface for run control and debug of the PowerPC Microprocessor though the parallel port of the PC. It translates software commands from various debugger vendors such as Cygnus, Green Hills, Metaware, Metrowerks, Microtec, Motorola, SDS, Tasking, and Wind River that utilize Macraigor's On-Chip Debug (OCD) Application Programming Interface (API). The Wiggler supports various Motorola microprocessors including CPU32, MPC5xx, MPC6xx, MPC7xx, MPC8xx, etc.

Macraigor provides several different but compatible products that perform this function at increasing levels of performance, functionality, and cost. The Wiggler is the least expensive, simplest device and provides an appropriate level of performance and capability for an inexpensive evaluation kit like Excimer, but for more serious software and hardware development the user may be interested in higher capability JTAG/COP interface tools from Macraigor or other vendors.

The third party debugger solutions provided by developers in the Excimer kit may utilize the Wiggler interface. Check their documentation.

Macraigor Systems provides a software application that tests and exercises the Wiggler and that can be used to control the processor, examine registers or memory, download code, etc. The Macraigor application is included in the Excimer kit and instructions for loading it onto WindowsNT or Windows95 are described in Section 1.2, "Installing the Wiggler Software."

1.2 Installing the Wiggler Software

To install Wiggler on Windows NT:

1. Log on as administrator.
2. Copy the file setupex.exe to a temporary directory.

3. Execute setupex.exe
4. Reboot the machine when the setup procedure is complete.
5. Log on as user (or remain as administrator).

To install Wiggler on Windows95:

1. No administrator activities are necessary.
2. Perform steps 2-5 above.

To use the wiggler on either NT or Windows95.

1. Connect the parallel port cable from the PC parallel port to the wiggler.
2. Connect the ribbon cable to the JTAG/COP header. (See caution above regarding Pin1.)
3. Copy the file, a:\wiggler\raysfiles\Wiggler.exe to your local directory.
4. Copy the files, a:\wiggler\cop_debugger* (i.e. all the files) to your local directory.
5. Using a command prompt window, execute Ocd_cmdt.exe. This is a more thorough test.
6. After using this test, terminate the test.

1.2.1 Using the Wiggler Software for Debug

Launch the Wiggler application. A window will appear with pull-down menus and control buttons to halt or reset the processor. A button displays the general purpose register (GPR) contents. A button allows the user to specify an address to “go to” and begin execution. A help command describes additional functionality. While this debugger is very basic, it is also very powerful and has several advanced capabilities, including icache and dcache display and modification and high-speed code download.

Part 2 Downloading a DINK Upgrade

DINK32 as supplied on the Excimer board is available from Motorola’s website in source or s-record format. While not officially supported, DINK32 serves as an application note of example PowerPC code. It is often enhanced with new functionality, support for new microprocessors or reference designs, and bug fixes. There may be enhancements or bug fixes after you have received your Excimer kit. Excimer can be easily upgraded by downloading the latest DINK32 as s-records into FLASHROM. The following paragraphs contain detailed instructions for downloading a new DINK32 using SmartCom and HyperTerminal.

To prepare to download:

1. Obtain the latest DINK32.src code from Motorola’s website. (Or download the source files and build your own.)
2. Reset Excimer and strike any key to interrupt the booting of DINK32 while in MDINK.

To download a new DINK32 from MDINK using **SmartCom**:

1. Put the flash into a cleared state by typing "fw -e" at the MDINK prompt. This will cause the flash to be erased and a new copy of MDINK will be copied from RAM (at 0x00000000) to the flash ROM (0xfff00000).

   WARNING: In X2 revision Excimer boards, the sector in FLASHROM where MDINK is stored is eraseable! Typing "fw -e" from the DINK32 prompt instead of MDINK will erase all of FLASHROM and the board will be inoperable at the next reset. Should such an unfortunate event occur, MDINK will have to be reloaded into flash through the JTAG port (probably in Motorola’s Apps Lab).

Revision X3 will have the hardware capability to protect the sector where MDINK resides and should be less vulnerable to unintentional erasure.

WARNING: The “fw -e” command takes a few minutes to execute. Do not power off the board or attempt to stop the “fw -e” command once it has begun. Since this command clears the flash and reloads MDINK, stopping the command will cause only a partial image of MDINK to be reloaded to ROM. Next time the board is reset, it will not function correctly.

2. DINK32 is a large file. Transfer time at 9600 baud will be lengthy. DINK allows the user to reset the baud rate to baud rates up to 57600. Check your terminal emulation package for the fastest rate supported. For example, if the fastest rate supported by your terminal emulation software is 57600, set the baud rate to 57600 by typing "sb -k 57600" in MDINK.
3. In the terminal emulator, change the baud rate to the new 57600 as well. Under Settings | Speed & Format, choose 57600 as the baud rate. Click OK in the dialog box and hit <return> in MDINK.
4. Change the autotype protocol so that there is no delay. Do this by selecting Settings | Autotype Protocol | Protocol Settings. In the space for "Delay", enter a 0 and click OK.
5. Disconnect communication from the board by pressing the telephone icon.
6. Under Connection | Choose Port | Flow Control, select the RTS/CTS option and click OK.
7. Reconnect to the board by pressing the telephone icon again.
8. Press <return> in MDINK.
9. Download the new dink32.src file:
   - Type "dl -fl -o ffc00000" at the MDINK prompt.
   - Hit the autotype icon in Smartcom. Use the directory tree to select the file to download and press the OK button.
   - The MDINK prompt returns when the download is complete.

     CAUTION: If communication or power is lost during this process, DINK32 in FLASHROM will be corrupted. If you can reset and get back to the MDINK prompt, be sure to start over at step one.
10. Set the baud rate in Smartcom back to 9600 (DINK defaults to 9600 baud). Choose 9600 in the baud rate field under "Settings | Speed & Format " and click OK.
11. Reset the board by pressing the reset button. MDINK will boot up and will branch to the DINK32 that was just downloaded.
12. If you wish to stop the download in step 10 after it has begun, type "S9" and hit <return>.
13. To return to MDINK from DINK32, type "go fff00100". MDINK will start again.
14. Steps 4 and 5 do not need to be repeated after the first time unless they are modified or the terminal emulation application is restarted.

To download a new DINK32 from MDINK using **HyperTerminal**:

1. Put the flash into a cleared state by typing "fw -e" at the MDINK prompt. This will cause the flash to be erased and a new copy of MDINK will be copied from RAM (at 0x00000000) to the flash ROM (0xfff00000).

   WARNING: In X2 revision Excimer boards, the sector in FLASHROM where MDINK is stored is eraseable! Typing "fw -e" from the DINK32 prompt instead of MDINK will erase all of FLASHROM and the board will be inoperable at the next reset. Should such an unfortunate event occur, MDINK will have to be reloaded into flash through the JTAG port (probably in Motorola’s Apps Lab).

Revision X3 will have the hardware capability to protect the sector where MDINK resides and should be less vulnerable to unintentional erasure.

WARNING: The “fw -e” command takes a few minutes to execute. Do not power off the board or attempt to stop the “fw -e” command once it has begun. Since this command clears the flash and reloads MDINK, stopping the command will cause only a partial image of MDINK to be reloaded to ROM. Next time the board is reset, it will not function correctly.

2. DINK32 is a large file. Transfer time at 9600 baud will be lengthy. DINK allows the user to reset the baud rate to baud rates up to 57600. Check your terminal emulation package for the fastest rate supported. For example, if the fastest rate supported by your terminal emulation software is 57600, set the baud rate to 57600 by typing "sb -k 57600" in MDINK.
3. Disconnect from the board by choosing Call | Disconnect.
4. Change the baud rate in HyperTerminal. Select File | Properties | Configure | Connect To and change the baud rate to 57600 in the space provided.
5. Reconnect to the board by selecting Call | Connect
6. Hit <return> in MDINK. You should return to the prompt.
7. Download the new dink32.src file:
   - Type "dl -fl -o ffc00000" at the MDINK prompt.
   - In HyperTerminal, select Transfer | Send Text File. Fill in the path to your DINK32 s-record in the space provided and press <return>. The file download should begin. If your computer locks up at this point, you will have to exit HyperTerminal, reset the board, and start over. At this point, only MDINK remains in the flash since the DINK32 image has been at least partially overwritten by the aborted dl attempt. You will have to perform this entire process again.
   - The MDINK prompt returns when the download is complete.

     CAUTION: If communication or power is lost during this process, DINK32 in FLASHROM will be corrupted. If you can reset and get back to the MDINK prompt, be sure to start over at step one.
8. Disconnect again by selecting Call | Disconnect.
9. Change the baud rate in HyperTerminal back to 9600 under File | Properties | Configure | Connect To and entering 9600 in the appropriate field.
10. Reset the board by pressing the reset button. MDINK will boot up and will branch to the DINK32 that was just downloaded.

Part 3 Excimer Memory Map

The EXCIMER memory map is shown in Table 2. Due to the partial hardware decode used on this simple design, unimplemented addresses generate no error signal and are aliased at multiple positions within the address range. The suggested address columns in Table 2 represent the address that DINK uses to address this function.

Table 2: Excimer Hardware Memory Map

Suggeste d Start	Suggested End	Address Start	Address End	R/W Size	Device	
0000 0000	00FF FFFF (007F FFFF in rev X2)	0000 0000	3FFF FFFF	1, 2, 4, 8 byte, burst 1, 2, 4, 8 byte	1MB of Static RAM	
4000 0000	4000 0000	4000 0000	4000 FFF0		COM2	Data register
4008 0000	4008 0000	4008 0000	4008 FFF0			Interrupt enable
4010 0000	4010 0000	4010 0000	4010 FFF0			FIFO control
4018 0000	4018 0000	4018 0000	4018 FFF0			Line control
4020 0000	4020 0000	4020 0000	4020 FFF0			Status LED (Modem control)
4028 0000	4028 0000	4028 0000	4028 FFF0			Line status
4030 0000	4030 0000	4030 0000	4030 FFF0			Modem status
4038 0000	4038 0000	4038 0000	4038 FFF0			Scratch
4040 0000	4040 0000	4040 0000	4040 FFF0		COM1	Data register
4048 0000	4048 0000	4048 0000	4048 FFF0			Interrupt enable
4050 0000	4050 0000	4050 0000	4050 FFF0			FIFO control
4058 0000	4058 0000	4058 0000	4058 FFF0			Line control
4060 0000	4060 0000	4060 0000	4060 FFF0			Error LED (Modem control)
4068 0000	4068 0000	4068 0000	4068 FFF0			Line status
4070 0000	4070 0000	4070 0000	4070 FFF0			Modem status
4078 0000	4078 0000	4078 0000	4078 FFF0			Scratch
		4079 0000	7FFF FFFF		Unused	
8000 0000	BFFF FFFF	8000 0000	BFFF FFFF		User/expansion I/O ($\overline{XCS1}$)	
FF80 0000	FFFF FFFF	C000 0000	FFFF FFFF	(1 read only), 2, 4, 8 bytes	Flash	

Part 4 DINK Memory Map

The hardware memory map shown in Table 2 is useful for programming input/output functions. For example, controlling the status LED in the coding example of section 1.1 on page 5 required knowledge of where the Serial One UART Control register could be addressed in the memory space. Equally important is the knowledge of where the user can download code without overwriting DINK or where functions or variables in DINK that are useful to the programmer are stored within memory.

Unlike the physical hardware of the UART addresses, the addresses DINK assigns to functions or variables may change with each compilation and link of DINK. The linker can be configure to generate a file which cross-references the symbol name of functions and global variables with their assigned physical address. A copy of this file name xref.txt is generally provided by Motorola with the s-record file for a new build of DINK. As an example, Table 3 provides some key addresses combed from the xref.txt for DINK32 version 10.3.

Table 3: DINK Memory Map

Use	Start	End
.text (program code)	0x00000000	0x00002bcb
.data	0x00024bd0	0x0003c192
.bss	0x0003c194	0x00041f1b
Stack	0x000f1f20	0x0005fffff
Reserved memory for DINK	0x00060000	0x0006fffff
Memory for User programs	0x00070000	0x000ffffff

Part 5 Using the Expansion Connector

Excimer provides a very limited expansion capability for input/output . The Berg connector on the board is an expansion connector that pins out the processor or FPGA signals shown in Table 1. See the Excimer schematics or the Minimal PowerPC System application note to understand this interface.

Table 4: Expansion Connector

Pin	Signal Name	Function
1, 2, 3	Vcc 3.3	Power
4, 6, 8, 10, 12, 12, 14, 16, 18	XD(0), XD(1), XD(2), XD(3), XD(4), XD(5), XD(6), XD(7)	Buffered Data Bus
5, 7, 9	Vcc 5.0	Power
11, 13, 15, 17	A(25), A(26), A(27), A(28)	Address Lines
19, 21, 23	NC	No Connect
20, 22, 24, 26, 28, 30, 32, 34	XF(0), XF(1), XF(2), XF(3), XF(4), XF(5), XF(6), XF(7)	Unused Pins of FPGA
36, 37, 38, 40	GND	Ground
39	XCLK 66MHz	Bus Clock
25	IRQ3*	Interrupt 3
27	IRQ2*	Interrupt 2
29	RESET*	Reset
31	XOE* ROE	Output Enable
33	XCS1* 0x8000000 0xBFFFFFF	Chip Enable
35	BWE*	Byte Write Enable

Part 6 Frequency Control

Motorola reserves the right to ship several different frequencies of PowerPC 603e processor on the Excimer board (Excimer often utilizes excess inventory). Consult the PowerPC 603e Hardware Specification on our website and your local Motorola sales office for more information on the range of frequencies that the 603e will operate at. The maximum frequency of operation under the full spec recommended temperature and voltage ranges is encoded in the part number on the top of the part. This number might be 233MHz or 266MHz or 300MHz (always integer or half integer multiples of the 66MHz Excimer clock speed). The part can be operated slower than this, to reduce power or for other reasons, as documented in the hardware specification. While operation above the maximum frequency or below the minimum frequency for the particular part is not recommended, the frequency can be changed via the PLL_CFG jumpers on the Excimer board. If the PLL_CFG on Excimer is configured for a frequency outside of the recommended frequency range for the processor installed, program execution may fail or be unreliable.

The PLL_CFG jumper settings for Excimer are shown in Table 5.

WARNING: The PLL_CFG jumpers should never be changed while power is applied to the processor.

CAUTION: The microprocessor, linear regulator, and heat sink area of the board (See Figure 2) can get very hot when the microprocessor is operated at high frequency (temperature at 300MHz is approximately 90°C/195°F) . When used for normal code development, 133MHz provides adequate performance and will reduce the danger of injury (temperature approximately 66°C/150°F) if this area of the board is inadvertently touched. If running above 133MHz, moving air from a nearby fan will assist in cooling the components; a clip-on heatsink for the microprocessor will further reduce the microprocessor's temperature.

Table 5. Excimer Frequency (PLL_CFG jumper) Settings

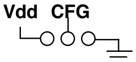
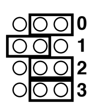
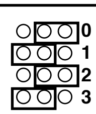
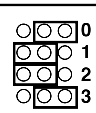
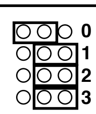
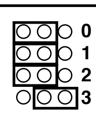
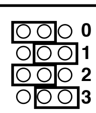
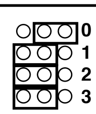
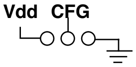
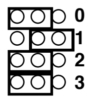
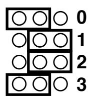
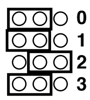
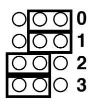
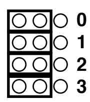
Jumper Position (seen as pin1 pin2 pin3) Vdd CFG 	PLL_CFG[0–3]	CPU Frequency in MHz (VCO Frequency in MHz)			Comment
		Bus-to-Core Multiplier	Core-to VCO Multiplier	Bus 66.67 MHz	
	0100	2x	2x	133	Part may not work here because of VCO limitation.
	0101	2x	4x	133	Should work here because VCO is 4x the core frequency.
	0110	2.5x	2x	166	
	1000	3x	2x	200	
	1110	3.5x	2x	233	
	1010	4x	2x	266	Check max processor frequency
	0111	4.5x	2x	300	Check max processor frequency

Table 5. Excimer Frequency (PLL_CFG jumper) Settings

Jumper Position (seen as pin1 pin2 pin3) Vdd CFG 	PLL_CFG[0–3]	CPU Frequency in MHz (VCO Frequency in MHz)			Comment
		Bus-to-Core Multiplier	Core-to VCO Multiplier	Bus 66.67 MHz	
	1011	5x	2x	333	Not recommended
	1001	5.5x	2x	366	Not recommended
	1101	6x	2x	400	Not recommended
	0011	PLL bypass		66	Not recommended
	1111	Clock off			Board will be inop

Part 7 Summary

The EXCIMER kit is a vehicle for exploring the PowerPC architecture and simple embedded software development. The kit is meant to be a low-cost example of a PowerPC design suitable for learning, teaching, or experimenting. Substantive applications that require large memory, PCI devices, or other advanced features should explore the Yellowknife platform also available from Motorola or evaluation systems from numerous third party vendors.

Appendix A Troubleshooting

Table 6 provides some failure conditions and probable causes.

Table 6. Failure Analysis

Problem	Symptom	Possible Cause
Nothing happens on the terminal after powerup.	All LEDs Off	5V Supply
	Error, Status, ROM LEDs on	ROM or RAM test failed
		Corrupted or no code in ROM
		PLL jumpers incorrect (too fast/slow)
		PLL in bypass (no jumpers installed)
		Power Supply Connection Faulty - unplug and reconnect
		Wiggler hardware interference - disconnect Wiggler and try again
	Error LED on, Status LED off	Duart test failed
dl command “hangs” forever	DINK prompt never returns	Corrupted s-records
		Baud rate too fast?
S-Record downloads cause DINK error 0xfb00..	Autotyping results in “unrecognized command or symbol error.”	Bits lost in serial transmission. Add delay between lines in Autotype protocol settings on terminal emulator.
Nothing happens on the terminal after go nnnnn command.	ROM access LED flashing	Code has branched into ROM and is looping there. Push reset.
	SRAM access LED flashing	Code is in an infinite loop in RAM. Wiggler might halt the processor and provide the location of the currently executing instruction.
	No access LEDs flashing	Code is in an infinite loop in cache or processor has taken a machine check. Push reset.
Performance less than expected	Code runs “slow” or SRAM access LED flashes during cache-bound program.	Caches disabled. Modify HID0 register to 0x8000c000
	Data accesses are slow	DCache marked “cache inhibited” for user code. Modify dbat1l register to 0x00000012.
Heat sink area or processor is “too hot” to touch.	Physical contact with the board is painful.	Processor is running “too fast” for ambient cooling conditions. Reduce speed via PLL_CFG jumpers or cool with moving air.

Motorola Literature Distribution Centers:

USA/EUROPE: Motorola Literature Distribution; P.O. Box 5405; Denver, Colorado 80217; Tel.: 1-800-441-2447 or 1-303-675-2140;
World Wide Web Address: http://ldc.nmd.com/

JAPAN: Nippon Motorola Ltd SPD, Strategic Planning Office 4-32-1, Nishi-Gotanda Shinagawa-ku, Tokyo 141, Japan Tel.: 81-3-5487-8488

ASIA/PACIFIC: Motorola Semiconductors H.K. Ltd Silicon Harbour Centre 2, Dai King Street Tai Po Industrial Estate Tai Po, New Territories, Hong Kong

Mfax™: RMFAX0@email.sps.mot.com; TOUCHTONE 1-602-244-6609; US & Canada ONLY (800) 774-1848;
World Wide Web Address: http://sps.motorola.com/mfax

INTERNET: http://motorola.com/sps

Technical Information: Motorola Inc. SPS Customer Support Center 1-800-521-6274; electronic mail address: crc@wmkmail.sps.mot.com.

Document Comments: FAX (512) 895-2638, Attn: RISC Applications Engineering.

World Wide Web Addresses: http://www.motorola.com/PowerPC/
http://www.motorola.com/netcomm/

MOTOROLA