

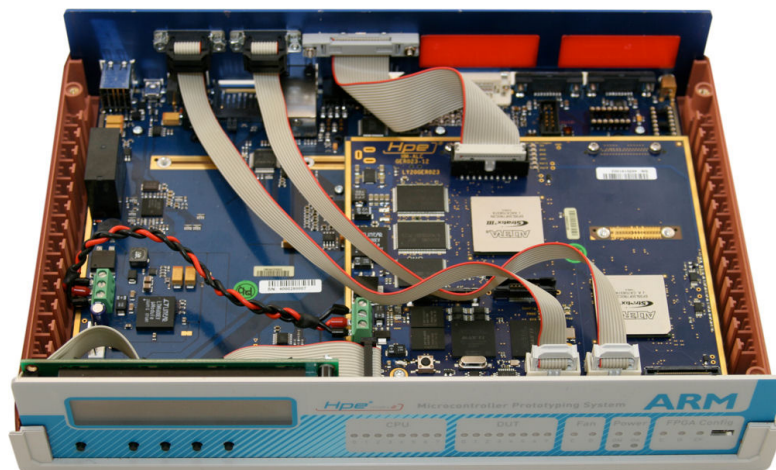
Using the Keil Microcontroller Prototyping System to develop an ARM Cortex-M class processor-based system

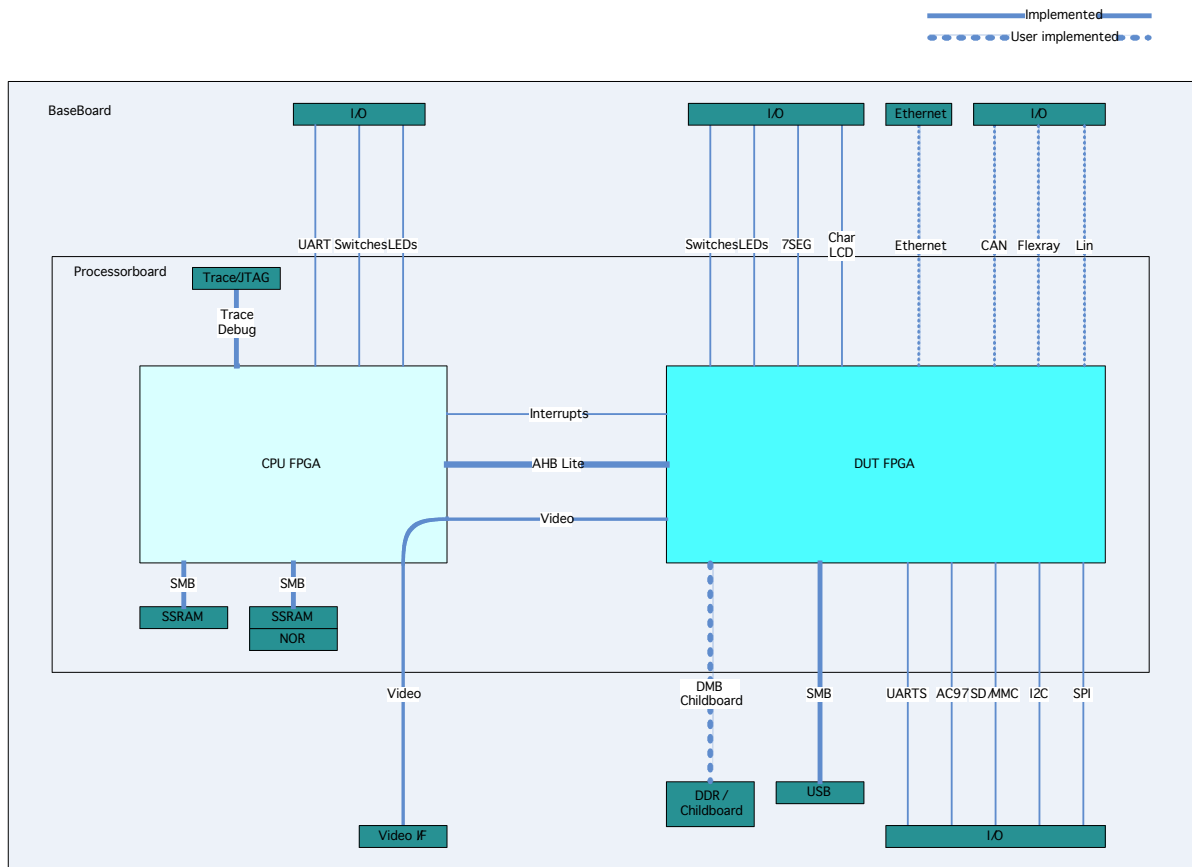
The increasing use of the ARM® Cortex™-M class processors in microcontrollers and embedded applications, together with the diversity of the processors within the family, requires a simple system to aid evaluation and prototyping. The Keil™ **Microcontroller Prototyping System** (MPS) allows early evaluation of Cortex-M class processors enabling selection of the correct processor for your application. The simple and flexible hardware development environment, combined with an integrated software environment to compile and debug your software, makes this the ideal solution from concept through to final silicon.



System Overview

The MPS is based around two Altera Stratix III EP3SL50 FPGAs. The first holds the processor and memory subsystem (CPU), including 64MB NOR Flash and 8MB SRAM; the second FPGA is for your hardware development (DUT). The processor FPGA is encrypted to allow a simple click-through license agreement (EULA) to be used for all supported processors. This means you can begin evaluating the selected processor quickly and without the need for access to the processor RTL. The CPU FPGA can also accept a non-encrypted image, so once you have licensed a processor you can make modifications and update the CPU FPGA with your own image.





MPB-M3 Block Diagram IO

To aid prototyping and allow you to start development work quickly, the MPS has an array of peripheral interfaces for you to use. Some are supported via the example system while others have just the physical interfaces designed to be used together with your own IP which would be programmed into the DUT FPGA. The list below details the peripherals and level of support.

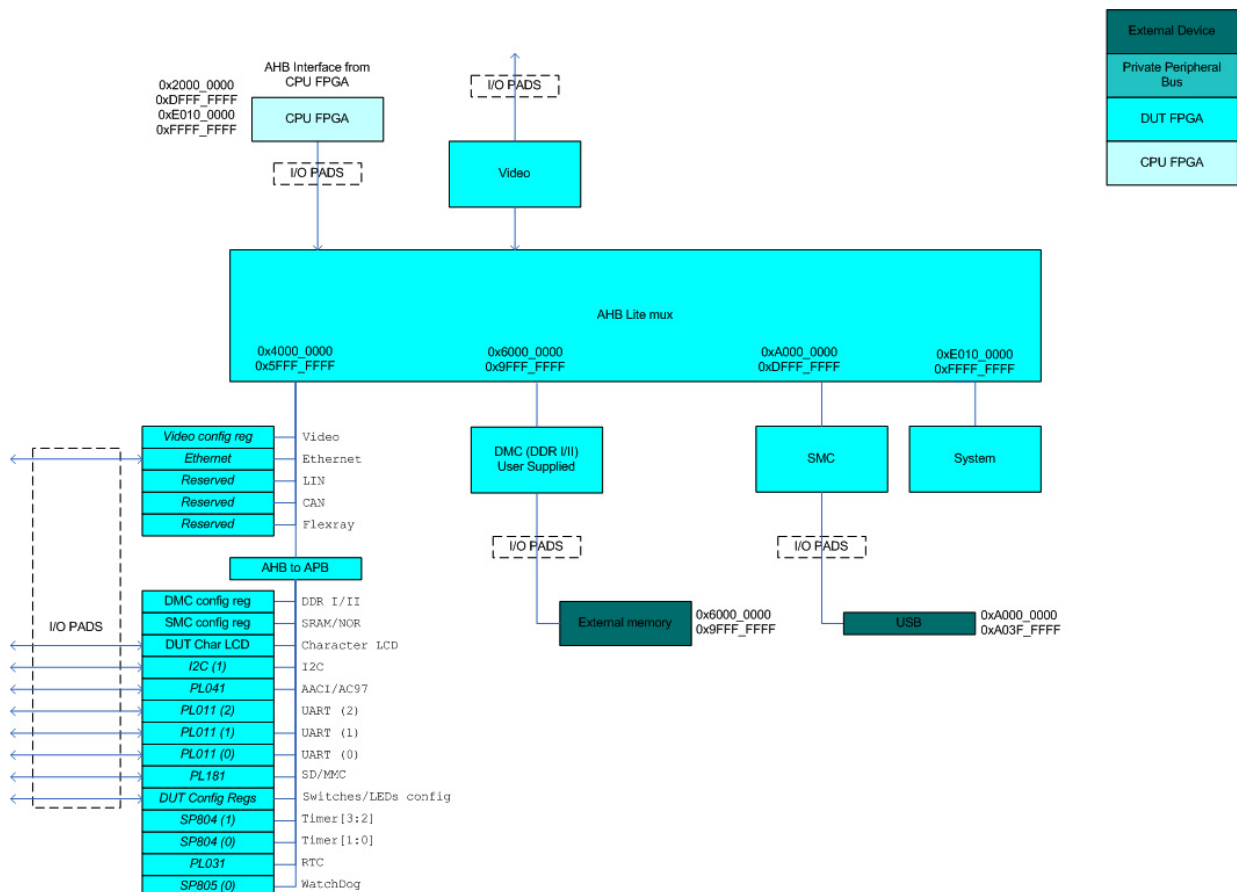
Peripheral	H/W Support	S/W Support	Note
Debug/Trace	ULINK2, RVI/RVT	MDK, RVDS	CPU FPGA
NOR Flash (32bit 64MB)	CPU FPGA	MDK, Boot Monitor	CPU FPGA
SRAM (32bit 8MB)	1 wait state CPU FPGA	MDK, Boot Monitor	CPU FPGA
UARTs	PL011 RTL	Selftest, Boot Monitor	DUT FPGA example
AC97	PL041 netlist	Selftest	DUT FPGA example
SD/MMCard	PL181 netlist	Boot Monitor	DUT FPGA example
Character LCD	RTL	Selftest, Boot Monitor	DUT FPGA example
LEDs/switches	RTL	Selftest, Boot Monitor	DUT FPGA example
USB 2.0 HOST/OTG	USB Chip	None	NXP ISP1761
Ethernet 10/100	Phy only	None	Requires MAC IP
CAN	Phy only	None	Requires IP
LIN	Phy only	None	Requires IP
FlexRay	Phy only	None	Requires IP
Video (up to XVGA)	Phy only	None	Requires IP

Table 1

In addition to the list of peripherals shown in Table 1, the DUT FPGA has a ‘Childboard’ interface to allow you to expand the I/O features of the system. This is a simple interface of FPGA I/O pins together with power pins enabling you to use ‘off the shelf’ boards or develop your own for custom requirements. A selection of ‘Childboards’ are available from Gleichmann Table 2 that enable you to add I/O functions to the MPS, ranging from simple I/O breakout boards with LEDs and headers to complex gigabit Ethernet Phys or DDR2 memories.

Childboard	Product number	Features
Connector LED board	HC-COLEv2	Five 26pin 0.1” pitch IDC headers. Each signal connected to LED and these can be turned on/off with jumpers
2 Channel Gbit Ethernet	HC-ETH2v2	Two National DP83865 phy’s with magnetics. MII, GMII and RGMII interface supporting 10/100/1000BASE-T
SDRAM	HC-SDRv2	256MB organized as 32M x64-bit at 120MHz
DDR2	HC-DDR	1GB of DDR2 RAM, product in development
NAND	HC-NAND4v2	Four 1GB devices with each with a separate 8-bit interface on the childboard connector
Synchronous SRAM and NOR Flash	HC-FRMEMv2	Two synchronous flow through NtRAM 1Mx32 (4MB) 8.5ns access time and two NOR Flash memories 16Mx32 (64MB) 70ns access time

Table 2: To accelerate your evaluation and development, an example system is shipped with the MPS for the DUT FPGA. This FPGA image supports most of the features on the board and is supplied with the example software and hardware design files necessary to rebuild both the software and hardware image for the FPGA.



We will now go through the design flow to rebuild, download and execute both the DUT FPGA image and software for execution on the MPS.

Hardware design

The example system is supplied as a mixture of RTL and netlists (for high value IP) allowing you to rebuild the FPGA image. The netlists for some IP blocks ensures a simple click-through license is all that is required. The top level architecture above the IP blocks is RTL based to enable easy modification.

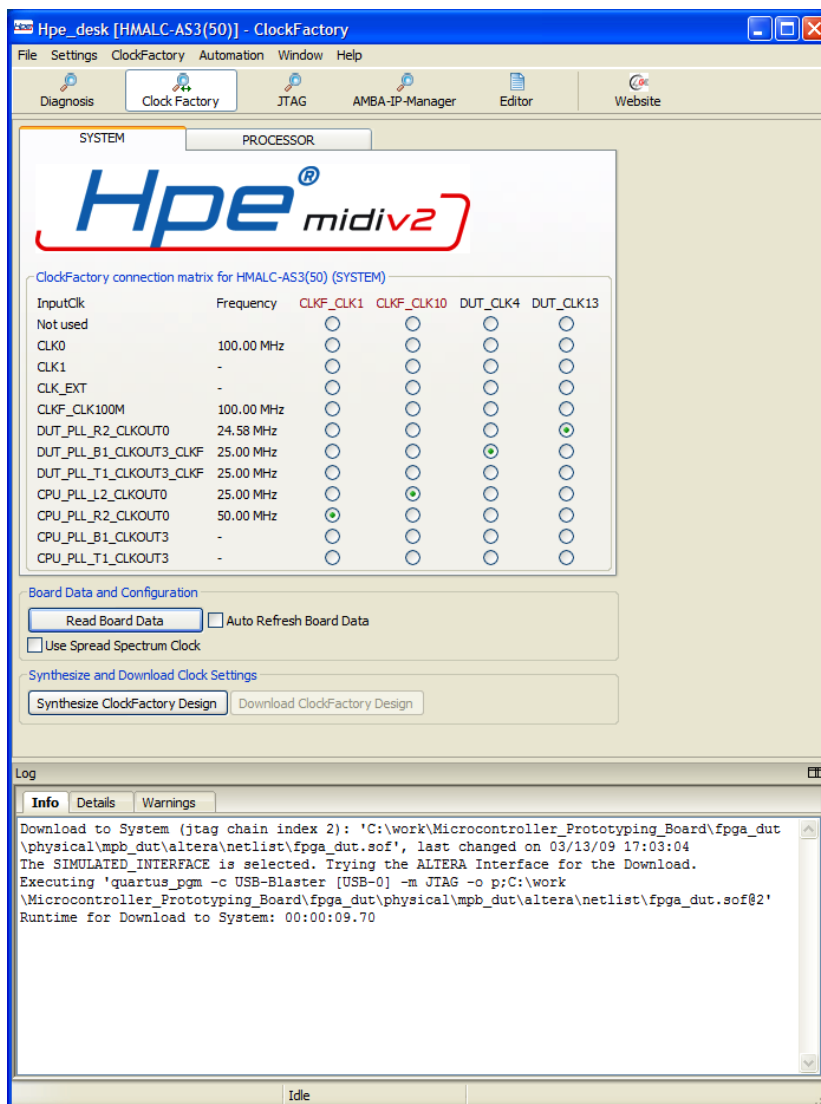
You need to edit and modify the RTL to add or remove features as you require. You can use the free Altera Quartus® II Web Edition tools to synthesize, and then place and route your final design in the FPGA. The example scripts used can easily be modified and contain the synthesis and timing constraints required to rebuild the FPGA image, and they can be used in the GUI or on the command line.

Downloading and configuring the MPS

You have now created the FPGA image as a .SOF file for the MPS. This can be downloaded into the MPS system via the HPE-Desk™ software supplied with the MPS. This is a Windows-based application that downloads images to the FPGA via the USB port on the MPS.

The HPE-Desk offers two options for downloading the image into the DUT FPGA. The first is 'download sof file to System FPGA' which is a fast method of downloading directly into the FPGA. However, this method is volatile meaning that when the system is rebooted, the FPGA is configured with the current system Flash image, any changes you have made to the FPGA image will be lost. This method is useful to save time during prototyping where you would like to quickly test small changes to the FPGA image.

The second option takes longer but is non-volatile as the image is downloaded to the system Flash. The 'download to System Flash Memory' option is ideal for making more permanent changes to the system image and should be used later in the development phase or to make 'milestone' changes to the system.

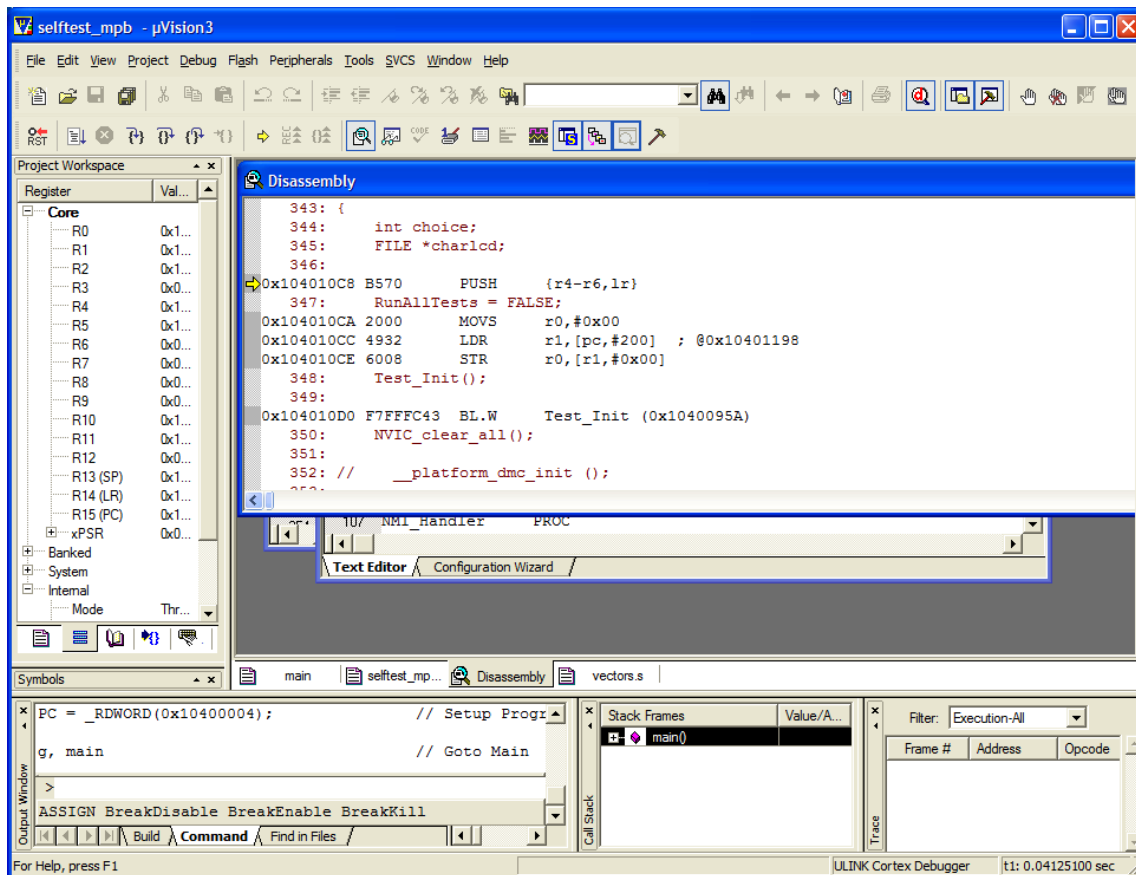


The second feature of the HPE-Desk is the ability to route different clock signal outputs from the FPGA back into the FPGA's clock inputs. This is basically a cross switch allowing you to drive clock sources (shown as rows) to clock inputs (shown as columns) in the FPGA. The default configuration is suitable for the example system, but if you wish to change the PLL in the DUT FPGA to create different clock frequencies for the processor and AHB lite interface then you can use this to switch between clocks without having to rebuild the FPGA images, saving time.

Software design

Once the hardware image is downloaded and working on the MPS you can begin developing and running a software application. This is a straight forward exercise as the MPS and its example system are fully supported by the Keil software development tools.

The MPS is supplied with an evaluation version of MDK-ARM (Microcontroller Development Kit) which is code size limited to 32KB, and the evaluation version can be upgraded to a full version of MDK-ARM by contacting Keil or our local distributor. The MPS also includes a ULINK2 adapter which connects your PC's USB port to your target system via JTAG or SWD, to download application code to the target and debug it.



Project Setup and Configuration

MDK-ARM contains a Device Database which allows you to easily setup and configure your project. Within the Device Database you only need to select 'ARM' and the required processor [Cortex-M3/M0] as the target device, and the required tool options and customized dialogs are automatically configured. As you continue your application development, only those options that are relevant to the selected device will be displayed, thus preventing selection of incompatible directives.

General Debug and Analysis

The μ Vision4 IDE/Debugger offers a wealth of debug and analysis tools to help you analyze and optimize your application code. In the Debugger, the Watch Window displays the values of automatic variables in the current function, the Memory Window displays various memory areas, and the Serial Window provides a terminal output for the on-chip UART. The Flexible Window Management System introduced in μ Vision4 enables developers to use multiple monitors and provides complete control over window placement anywhere on the visual surface. μ Vision4 includes a Logic Analyzer which records the values of variables and peripheral I/O signals over time and displays them. These signal values can be displayed in three different formats: bits, showing outputs as a logic level 0 or 1; State, showing stage changes of a value; and Analog, showing a graphical representation.

Cortex-M class processor Debug & Trace

MDK-ARM and ULINK2 have extended features for debugging applications running on Cortex-M3 and Cortex-M0 processor-based systems using the standard JTAG, or the advanced Serial-Wire debug modes. Additionally, Data Trace Windows provide information from a running Cortex-M3 processor-based system for program data, exceptions, variables and printf-style outputs. For example, you may use the Trace Records window to view all the trace records captured during the debug session, including overflows and timestamps; while the Exception Trace windows show which exceptions that have taken place, including the number of times they have occurred and how long the system spent handling the exception. You may also view *printf*-style debug or other program-specific information using the ITM Viewer window or keep a track of the events in your application using the Event Counter window.

Summary

The Microcontroller Prototyping System provides a full solution for developing and evaluating Cortex-M processor-based systems. It is available now via Keil and our worldwide distribution channels. Further information is available at www.keil.com/mps