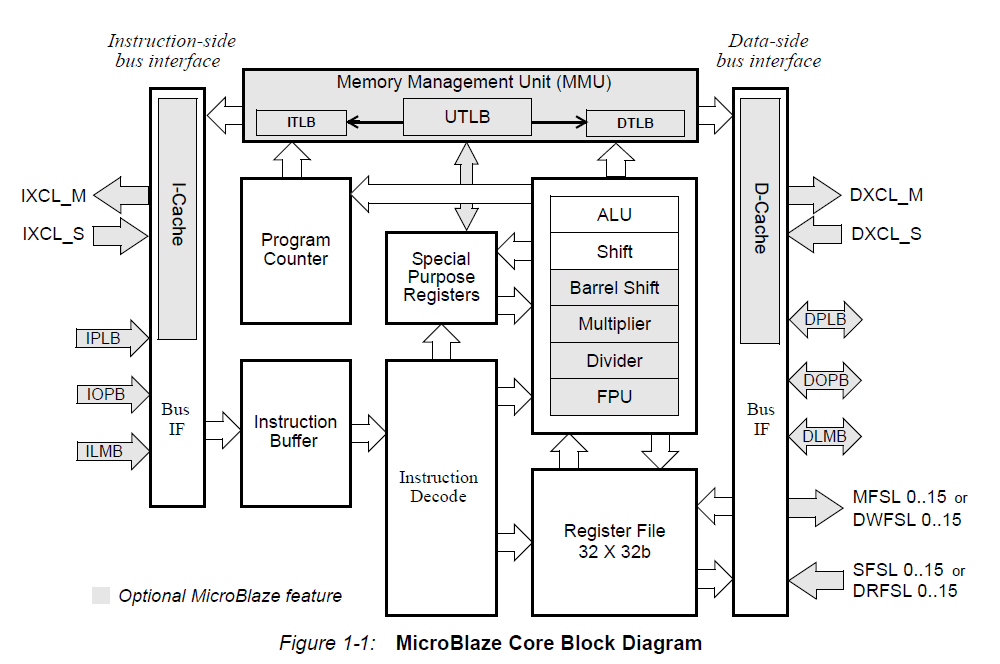
# MicroBlaze Overview

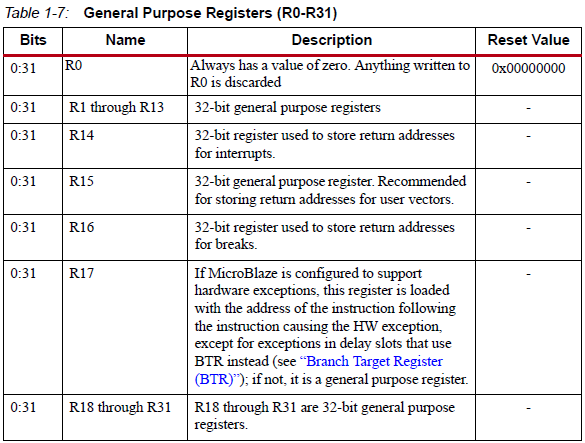
Microblaze soft processor ip is reduced instruction set computer(RISC) optimized for implementation in Xilinx FPGAs. Block Diagram of MicroBlaze is shown below.



Gray parts on the diagram are optional parts and they can be added or removed by configuring the IP in the Vivado project. Others are fixed and cannot be configured.

Microblaze uses Big-Endian format to represent data and the supported data types are word, half word, and byte.

Microblaze has 32 32-bit general purpose registers and up to 18 32-bit special purpose registers, depending on configurations.



|  |  |
| --- | --- |
| Special Purpose Registers | |
| PC | Program Counter |
| MSR | Machine Status Register |
| EAR | Exception Address Register |
| ESR | Exception Status Register |
| BTR | Branch Target Register |
| FSR | Floating Point Status Register |
| EDR | Exception Data Register |
| PID | Process Identifier Register |
| ZPR | Zone Protection Register |
| TLBLO | Translation Look-Aside Buffer Low Register |
| TLBHI | Translation Look-Aside Buffer High Register |
| TLBX | Translation Look-Aside Buffer Index Register |
| TLBSX | Translation Look-Aside Buffer Search Index Register |
| PVR | Process Version Register |

MicroBlaze Instructions are pipelined. For most instructions, each stage takes one clock cycle to complete. Number of pipeline stages can be configured by setting C\_AREA\_OPTIMIZED. To minimize the hardware cost three stage pipeline can be used by setting C\_AREA\_OPTIMIZED 1. And to maximize the performance five stage pipeline can be used by setting C\_AREA\_OPTIMIZED 0.

The MicroBlaze cores are organized as a Harvard architecture with separate bus interface units for data and instruction accesses. Supported bus interfaces are listed below.

* Local Memory Bus (LMB) : Provides single-cycle access to dual port block RAM.
* Processor Local Bus (PLB) or On-Chip Peripheral Bus (OPB) : Provide a connection to both on-chip and off-chip peripherals and memory.
* Xilinx CacheLink (XCL) : provides interface between caches and external memory controllers.

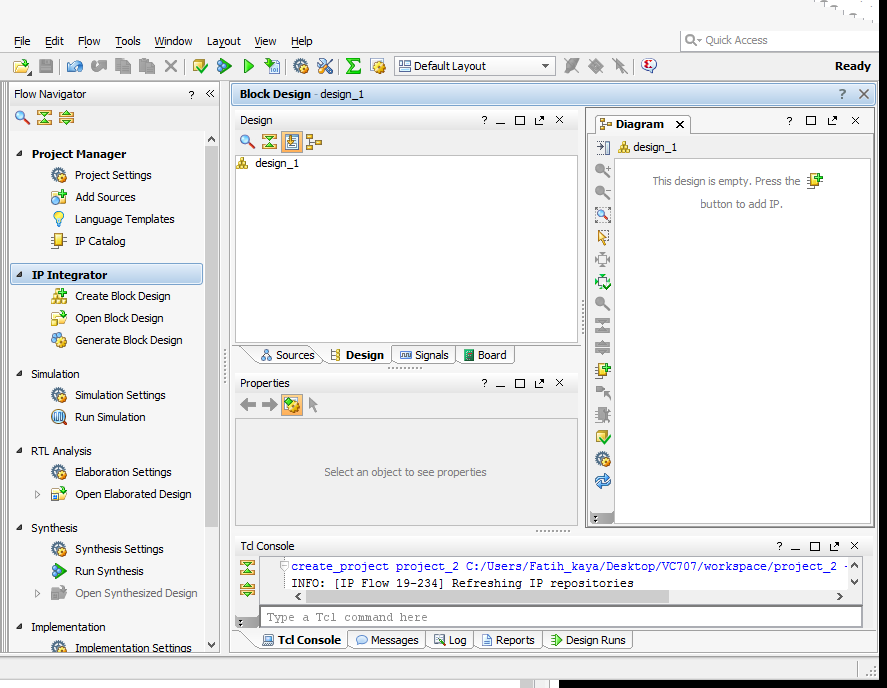
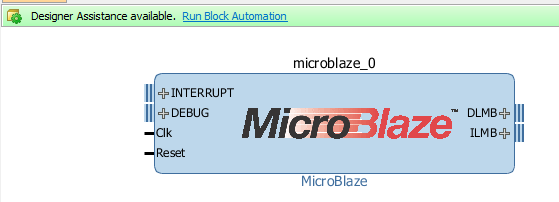
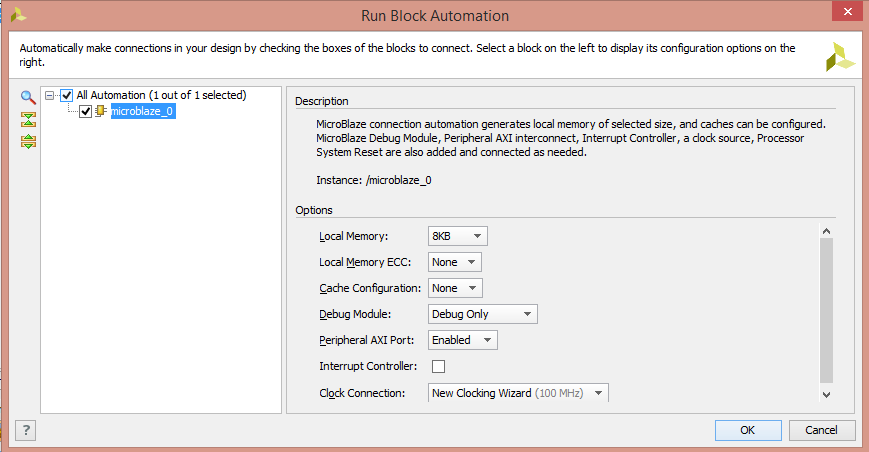
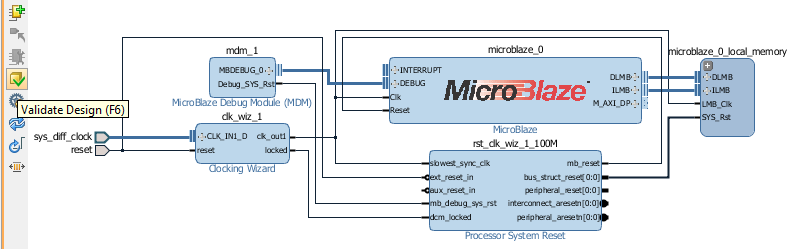
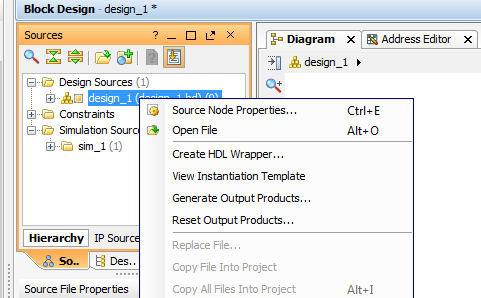
There is some IP block not mandatory to add to the project to work with MicroBlaze. But they are required to set up a basic MicroBlaze system. They are LMB Block RAM, MicroBlaze Debug Module (MDM), Processor System Reset, and Clocking Wizard.

* LMB Block RAM: Provides a low area, high frequency and low latency connection for the MicroBlaze DLMB (for Data) and ILMB (for instructions) ports to device block RAM. The supported block RAM sizes are determined by the Block Memory Generator (4 – 256 KB).
* MDM: Enables JTAG-based debugging of one or more (up to 32) MicroBlaze processors. Provides a configurable AXI4 Master port for direct access to memory from JTAG.
* Clocking Wizard: Accepts up to two input clocks and up to seven output clocks per clock network. Provides feature to monitor the clocks. Optionally buffers clock signals.
* System Processor Reset: Reset inputs are selectable as active-High and active-Low.

# How to Open a MicroBlaze Project in Vivado

Xilinx has two software to design, synthesize and implement hardware for their FPGAs. Xilinx ISE which is the older software does not support newer FPGA boards. VC707 evaluation board is one of them so Vivado which is newer software, is used to design MicroBlaze project on VC707.To open a new Microblaze following steps are done.

## Hardware Design Part

* Select “Create New Project” from quick start or File menu.
* Set “Project Name” and “Project Directory” then press “Next”.
* Select “RTL Project” on the opening page then press “Next”.
* Select the chip or board on the opening page. Board selection is recommended for easy pin assignment.
* After all, select finish. Then project will open.
* Select “Create Block Design” below “Project Manager/IP Integrator” menu. And name the design.
* This will create a new block design project. After all, window must look as following.
* Click Screen Clipping symbol to add a new IP. Then search and select Microblaze on the opening window.
* Click on the “Run Block Automation”. This wizard will add necessary blocks to your design and makes their connection depending on your selection
* 
* Local Memory can be increased up to 128 KB if the program written in c needs more memory. Memory requirement error can be faced in large software. Also, cache can be added up to 64 KB to increase performance. Clock connection can be selected any clock pin.
* After configuration done. Click “OK”. And click “Run Connection Automation”. Screen Clipping Select all connection configurations. That will make the necessary connection.
* 
* Your design must seem as above. Click Validate Design to check whether there is any error in the connections.
* On the Sources/Hierarchy tab right click on the design and select “Create HDL Wrapper”.
* Now it is ready to generate Bitstream. Click on “Generate Bitstream”. Select “Yes” and “Ok” to all opening windows.
* After Bitstream is generated successfully select “File/Export/Export Hardware”
* Tick the “Include Bitstream” selection. Then click Ok.
* Select “File/Launch SDK” to open SDK where C code to load to designed microprocessor is written.

## Coding Part

After SDK is opened the hardware platform folder is generated which is named as <Your Design Wrapper Name>\_hw\_platform\_0 and includes three files with .bit, .mmi and .hdf extensions. First thing we need to the is generating Board Support Package (BSP).

* Select “File/New/ Board Support Package” to generate a new BSD file. Name it and click finish.
* Select “File/New/ Application Project”.
  + Name the Project
  + OS platform = standalone
  + Hardware Platform = <Your Design Wrapper Name>\_hw\_platform\_0
  + Processor = microblaze\_x
  + Language = Choose which you want to write
  + Board Support Package = Can be created one more or can be used existing one. Select the use existing one. (It depends on you.)
  + Click finish or click next and select an example project.

|  |  |
| --- | --- |
| |  | | --- | | **Name of BSP Folder**  **BSP Documentation**: Includes documents for libraries of the IPs. These documents contain nothing but the comments of the functions in the driver libraries at “microblaze\_0/libsrc/.”.  Documentation for Block Ram IP  **libsrc** : Includes Driver libraries and example projects for IPs. All IP controllable IP block must be seen here. |   Screen Clipping |

Up to now only simple MicroBlaze system is designed. From now new peripherals will be added and controlled on software.