POWERPC BASED EMBEDDED DESIGN IN FPGA USING XILINX EDK
About PowerPC 405

- 32-bit instruction/64-bit data
- Pipeline:
  - 5-stage pipeline, mostly single cycle but for multiply(4 cycles)/division(35 cycles)
- Memory addressing
  - Support unaligned load/store
  - Little endian operation
- Cache:
  - 16K, 2-way cache, 32 byte block size
- Support for on-chip memory (OCM)
  - With performance identical to a cache hit
- Buses:
  - PLB, OPB, DCR, OCM
PowerPC-based Embedded Design

Full system customization to meet performance, functionality, and cost goals
PPC Buses

- **Processor Local Bus (PLB)**
  - One 32-bit address/three 64-bit data buses attached to the instruction-cache and data-cache units.
    - Two of the 64-bit buses are attached to the data-cache unit, for read/write operations. The third 64-bit bus is attached to the instruction-cache unit to support instruction fetching.
    - To provide a high-bandwidth, low-latency connection between bus agents that are the main producers and consumers of the bus transaction traffic.
      - connect your higher speed peripherals (e.g., G-Ethernet Mac) and memory.

- **On-chip Peripheral Bus (OPB)**
- **Device Control Register (DCR)**
- **On-chip Memory Bus (OCM)**
PPC Buses

- Processor Local Bus (PLB)
- **On-chip Peripheral Bus (OPB)**
  - A fully synchronous 32-bit address and 32-bit data bus.
  - To provide a flexible connection path to peripherals and memory, while providing minimal performance impact to the PLB bus
    - put slower peripherals on this bus, such as UARTs, GPIO, 10/100 E-Net MAC, etc.
- Device Control Register (DCR)
- On-chip Memory Bus (OCM)
PPC Buses

- Processor Local Bus (PLB)
- On-chip Peripheral Bus (OPB)
- Device Control Register (DCR)
  - A 32-bit bus for accessing device control registers
  - Most traffic occurs during the system initialization period; however, some elements, such as the DMA controller and the interrupt controller cores, use the DCR bus to access normal functional registers used during operation.
- On-chip Memory Bus (OCM)
User Defined DCR

- User Defined DCR Bus
  - A 10-bit address bus.
  - Separate 32-bit input data and output data busses.
  - Separate read and write control signals.
  - A read/write acknowledgement signal.

- Chained DCR Peripheral Devices

- Using `mfdcr` and `mtdcr` to access DCRs
  - `mfdcr`: move from a special dcr
  - `mtdcr`: move to a special dcr
PPC Buses

- Processor Local Bus (PLB)
- On-chip Peripheral Bus (OPB)
- Device Control Register (DCR)
- On-chip Memory Bus (OCM)
  - Separate bus interface for non cacheable memory
  - Separate I-side and D-side OCM controllers inside the processor block for higher performance
OCM Bus

- **Features**
  - Independent 16-MB logical space for each of the DSOCM and ISOCM
    - 16 MB must be reserved regardless of actual memory used
  - ISOCM bus: 64-bit for instruction fetches and 32-bit write to initialize or test
  - DSOCM bus: 32-bit data-read and 32 bit data-write
  - Up to 128 KB / 64 KB (ISOCM / DSOCM) using programmable BRAM aspect ratios
    - Programmable processor versus BRAM clock ratio
OCM Bus

- **Benefits**
  - Avoids loads into cache, reducing pollution and thrashing
  - Has fast-fixed latency of execution
  - On the D-side, dual-port BRAM enables a bidirectional data connection with the processor

- **Sample uses**
  - I-side: Interrupt service routines, boot-code storage
  - D-side: Scratch-pad memory, bidirectional data transfer
# Bus Timing

<table>
<thead>
<tr>
<th>Transaction synchronous with</th>
<th>PLB CLK</th>
<th>OPB CLK</th>
<th>DCR CLK</th>
<th>OCM CLK *</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor clock</td>
<td>Processor clock</td>
<td>PLB clock</td>
<td>Processor clock</td>
<td>Processor clock</td>
</tr>
<tr>
<td>Clock ratio</td>
<td>1:1 to 16:1</td>
<td>1:1 to 4:1</td>
<td>1:1 to 8:1</td>
<td>1:1 to 4:1</td>
</tr>
<tr>
<td>Example</td>
<td>Processor clock at 300 MHz, PLB at 100 MHz</td>
<td>PLB at 100 MHz, OPB at 50 MHz</td>
<td>Processor clock at 300 MHz, DCR at 100 MHz</td>
<td>Processor clock at 300 MHz, OCM at 150 MHz</td>
</tr>
</tbody>
</table>

- Use timing constraints to determine which ratio to use
- *There are two independent clocks for each OCM controller:
  - BRAMDSOCMCLK
  - BRAMISOCMCLK

[PowerPC™]
About Embedded Development Kit

- The Xilinx software suite for designing complete embedded programmable systems

- Tools, documentation, and IP for designing systems with embedded IBM PowerPC™ hard processor cores, and/or Xilinx MicroBlaze™ soft processor cores

- Integrated design environment for both hardware and software components
Embedded Design in an FPGA

- Embedded design in an FPGA consists of the following:
  - FPGA hardware design
  - C drivers for hardware
  - Software design
    - Software routines
    - Interrupt service routines (optional)
    - Real Time Operating System (RTOS) (optional)
Embedded Development Overview

**Standard Embedded SW Development Flow**
- Code Entry
- C/C++ Cross Compiler
- Linker
- Load Software Into FLASH
- Debugger

**Standard FPGA HW Development Flow**
- VHDL or Verilog
- HDL Entry
- Simulation/Synthesis
- Implementation
- Download Bitstream Into FPGA
- Chipscope

**Compiled ELF**
- Embedded Development Kit
  - Board Support Package
  - System Netlist
  - Data2MEM
  - Download Combined Image to FPGA

**Compiled BIT**
- RTOS, Board Support Package
- Download Bitstream Into FPGA
- Chipscope
Xilinx Platform Studio

- Xilinx Platform Studio (XPS) is a graphical Integrated Design Environment (IDE) for hardware/software design and verification
  - Project management
  - Platform management
  - Software application
Xilinx Platform Studio (XPS)
Hardware Creation

- **Platform Generator – PlatGen**
  - **Input file** → Microprocessor Hardware Specification file and Microprocessor Peripheral Description file (MPD)
    - MHS file defines the configuration of the embedded processor system including bus architecture, peripherals and processor(s), interrupt request priorities, and address space
    - MPD file defines the configurable parameters with their default values and available ports for a peripheral
  - **Output files** → system netlist, peripheral netlists, and Block Memory Map (BMM) file
    - BMM describes how individual block memory map constitutes the continuous logical memory space
Hardware Implementation

- Hardware netlists must be implemented with the Xilinx implementation tools
- Either the ISE™ Project Navigator or Xflow batch tool can be used to implement the design
  - The ISE Project Navigator GUI gives you access to the whole suite of Xilinx design entry and physical implementation point tools
  - Xflow is a non-graphical tool that encapsulates the Xilinx implementation/simulation flows and enable an simplified flow to implement the hardware and generate the bitstream
Software Library Generation

- Library Generator – LibGen
  - Input files → Microprocessor Software Specification (MSS)
  - Output files → libc.a, libXil.a, libm.a
  - LibGen is generally the first tool run to configure libraries and device drivers
    - The MSS file defines the drivers associated with peripherals, standard input/output devices, interrupt handler routines, and other related software features
  - LibGen configures libraries and drivers with this information and produces an archive of object files:
    - libc.a - Standard C library
    - libXil.a - Xilinx library
    - libm.a - Math functions library
Software Compilation

- Compile program sources
  - Input files → *.c, *.c++, *.h, libc.a, libXil.a, libm.a
  - Output files → executable.elf
  - This invokes the compiler for each software application and builds the executable and linkable format (ELF) files for each processor
    - Executable and Linkable Format (ELF)
      - a common standard file format for executables, object code, shared libraries, and core dumps
      - Flexible and extensible
      - Not bound to any particular processor
Merging Hardware and Software Flows

Hardware Flow

data2MEM

download.bit

Software Flow

MicroBlaze™, PPC, Arbiter, UART, GPIO
Merging Hardware and Software Flows

- Data2MEM – Update the bitstream
  - Input files → `system_bd.bmm`, `system.bit`, `executable.elf`
  - Output file → `download.bit`
  - This invokes the BitInit tool, which initializes the instruction memory of the processor
  - This is the stage where the hardware and the software flows come together. This stage also calls the hardware and software flow tools if required
Configuring the FPGA

- Download the bitstream
  - Input file → *download.bit*
  - This downloads the *download.bit* file onto the target board using the Xilinx iMPACT tool in batch mode
  - XPS uses the *etc/download.cmd* file for downloading the bitstream
    - The *download.cmd* file contains information such as the type of cable is used and the position of the FPGA in a JTAG chain
About IPIF
Summary

- PowerPC architecture
- Embedded development flow
- Xilinx EDK and the development flow