INTRODUCTION TO EMBEDDED SYSTEMS

5th Detector Workshop of the Helmholtz Alliance “Physics at the Terascale”
14 – 16 March 2012, Physikalisches Institut Universität Bonn
FPGA School

Bonn 14th March 2012

Tomasz Hemperek

PHYSICS AT THE TERA SCALE
Helmholtz Alliance

universitätbonn

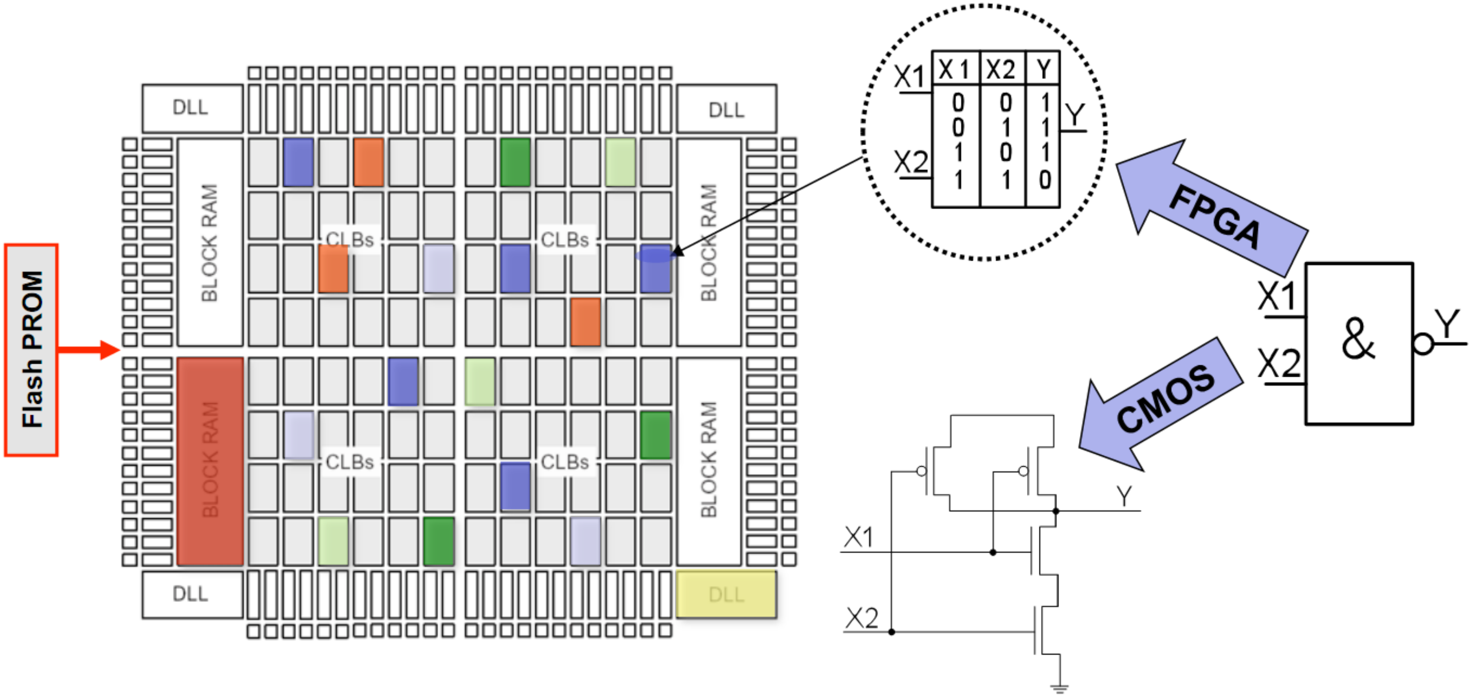
SI LAB
Silizium Labor Bonn

Outline

- FPGA
- Computer Architecture in 5 min
- Embedded systems
- Bus as a interconnect
- Microblaze ecosystem
- EDK introduction
- SDK introduction

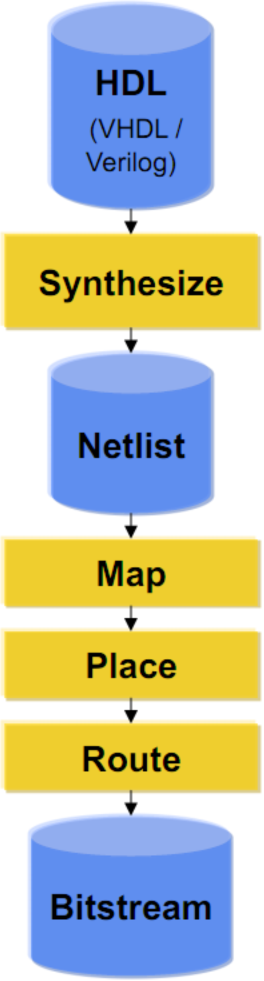
Field-Programmable Gate Arrays (FPGAs)

- Fine-grained reconfigurable hardware
- Gate-Array: regular structure of “logic cells”, connected through an interconnection network
- Configuration stored in SRAM, must be loaded on



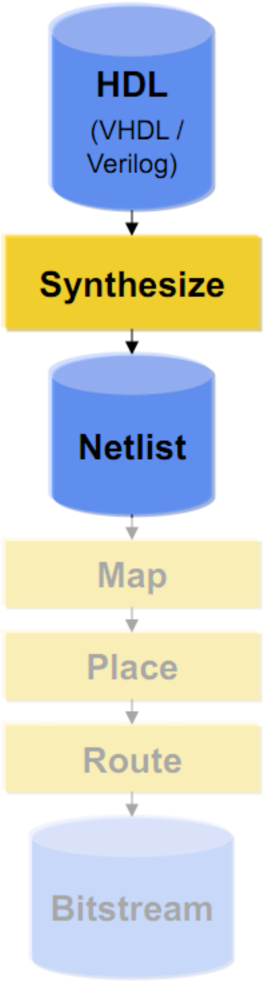
X 1	X2	Y
0	0	1
0	1	1
1	0	1
1	1	0

FPGA toolflow

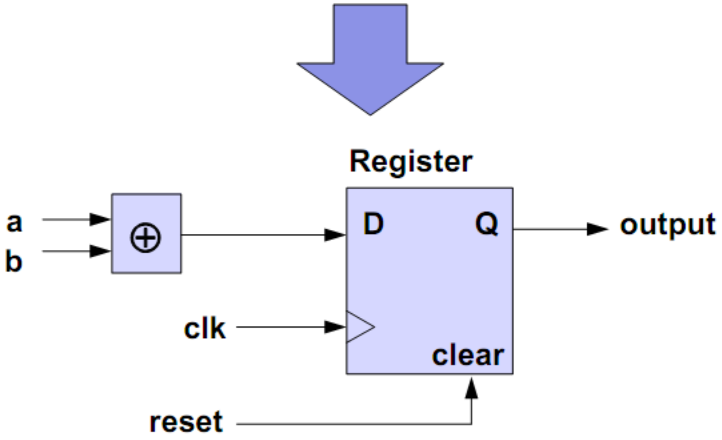


- Hardware design is traditionally done by modeling the system in a hardware description language
- An FPGA “compiler” (synthesis tool) generates a netlist
- which is then mapped to the FPGA technology
- The inferred components are placed on the chip
- The connecting signals are routed through the interconnection network

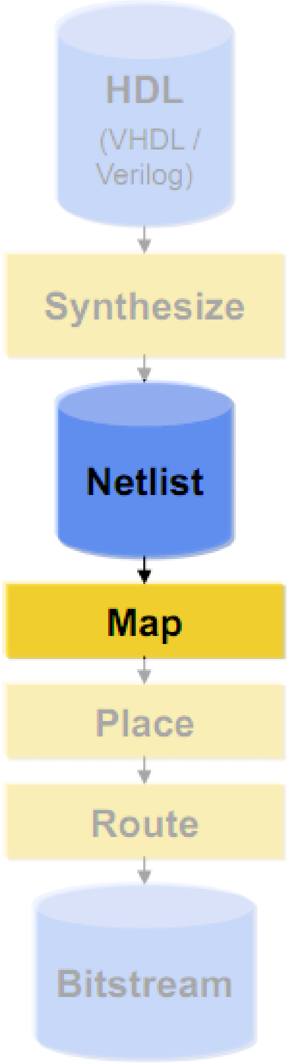
HDL Synthesis

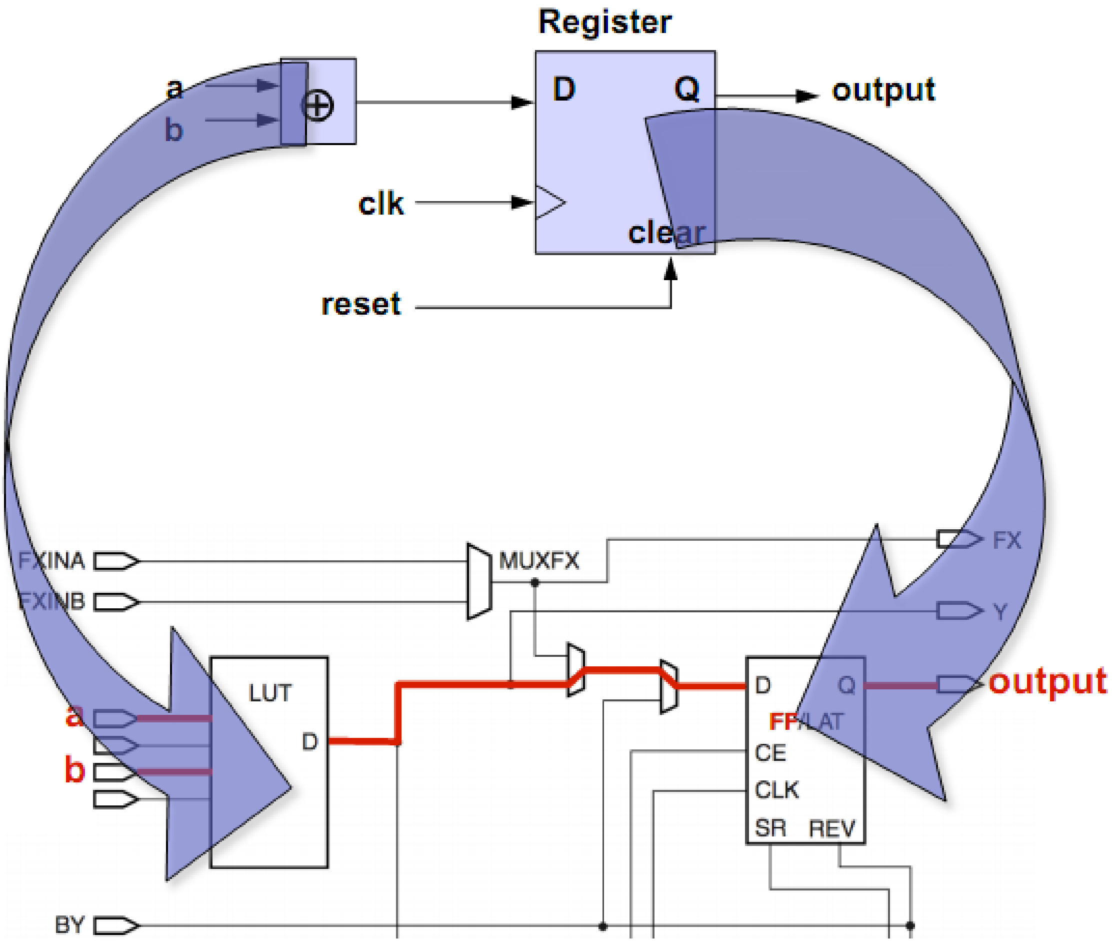


```
process(clk, reset)
begin
  if reset = ‘1‘ then
    output <= ‘0‘;
  elsif clk‘event AND clk = ‘1‘ then
    output <= a XOR b;
  end if;
end process;
```

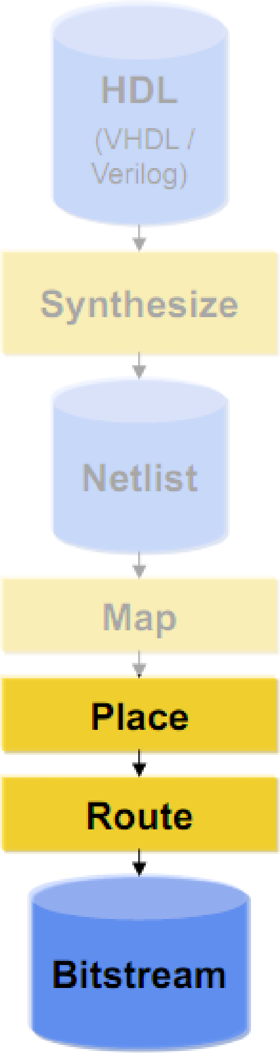


Technology Mapping





Place & Route



HDL (VHDL / Verilog)

Synthesize

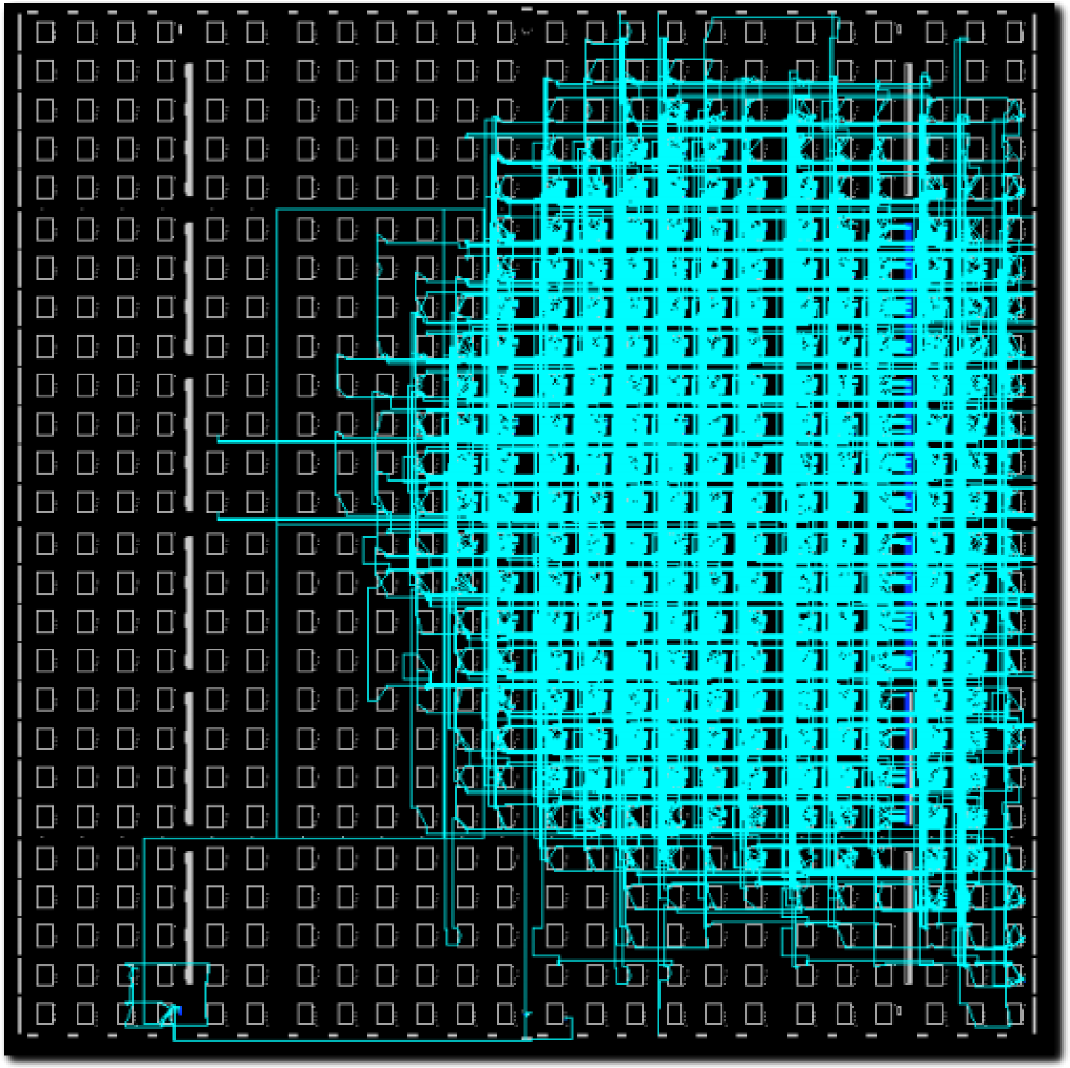
Netlist

Map

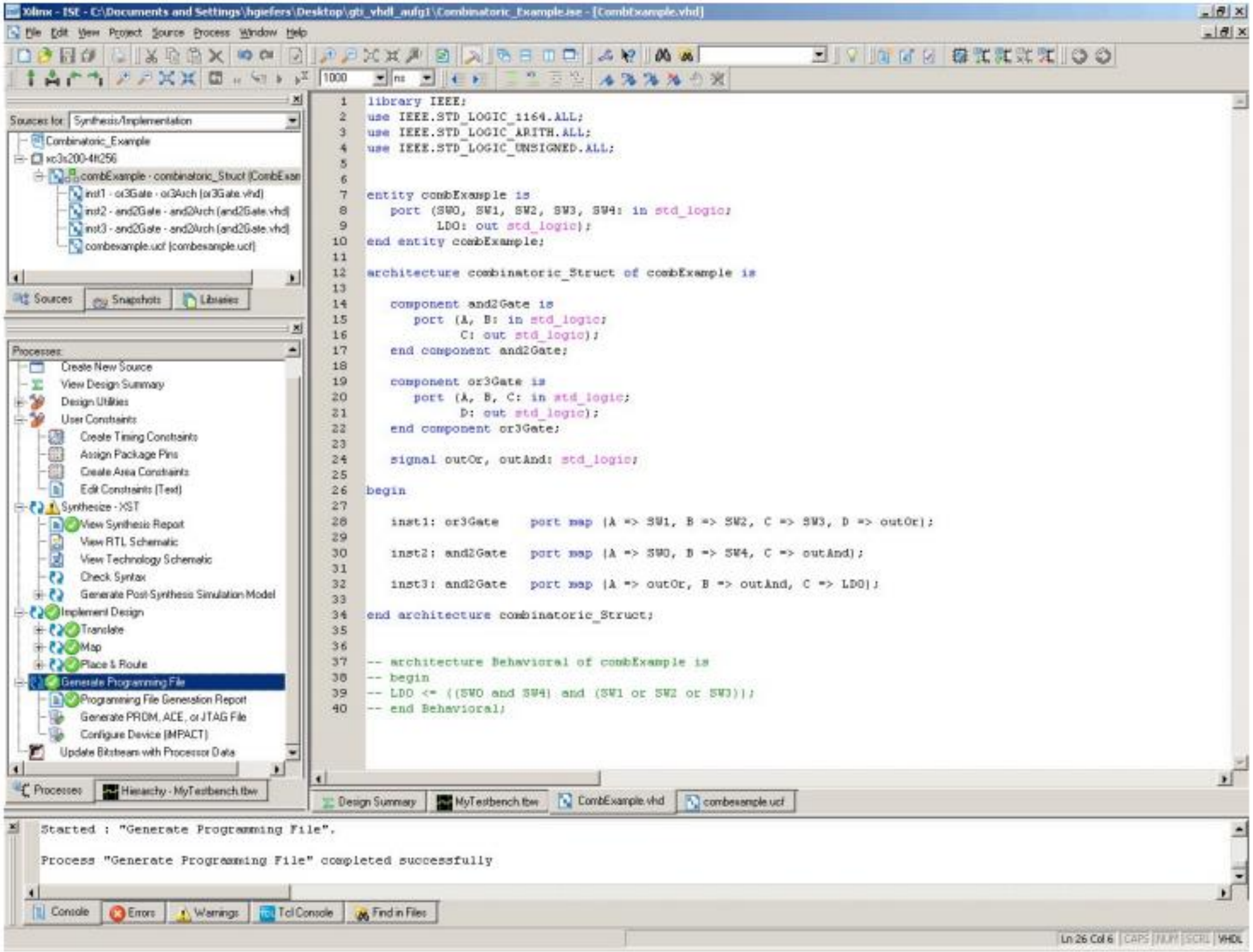
Place

Route

Bitstream



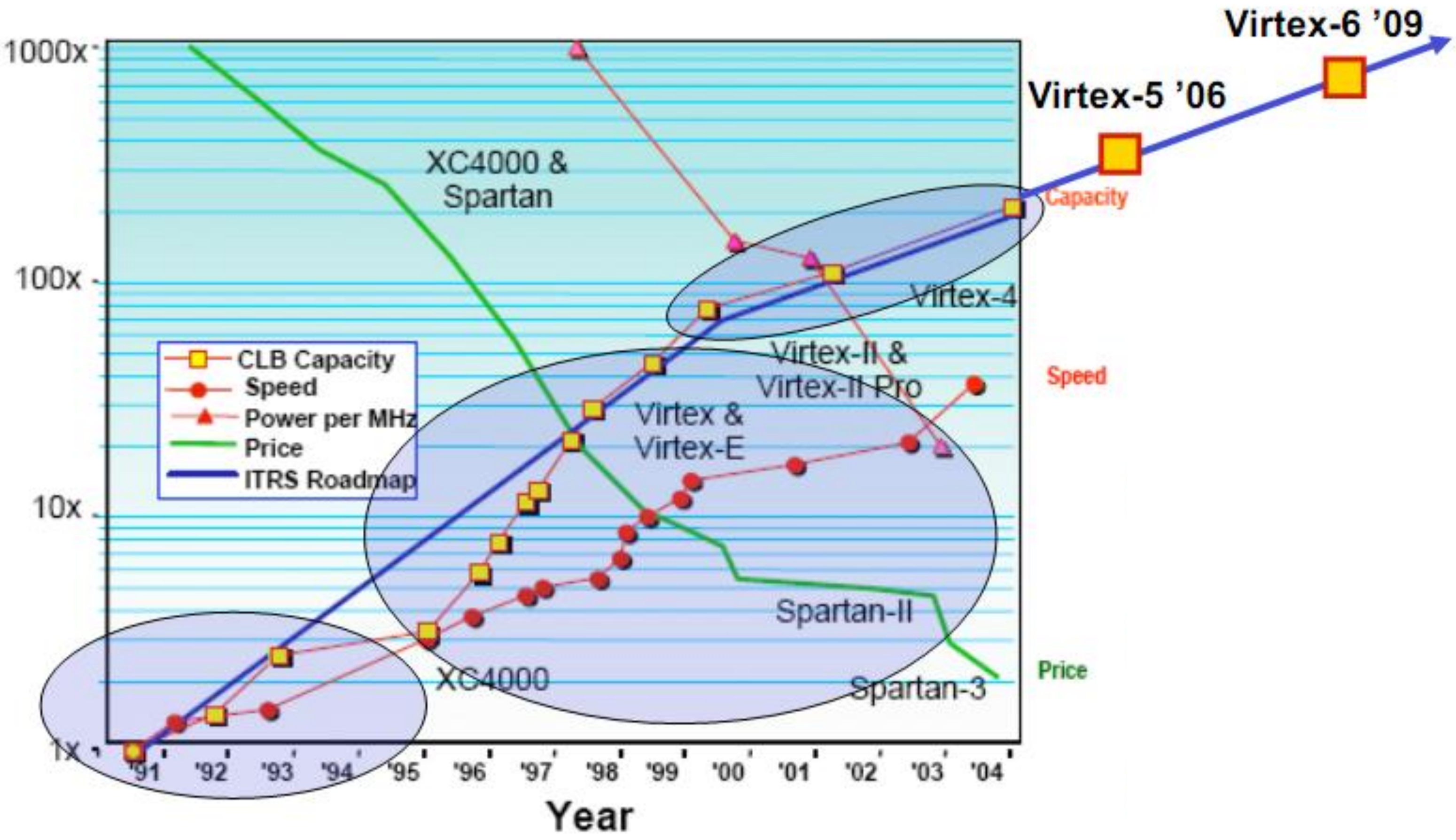
Xilinx ISE


Xilinx - ISE - C:\Documents and Settings\hgiefers\Desktop\gti_vhdl_aufg1\Combinatoric_Example.ise - [CombExample.vhd]
File Edit View Project Source Process Window Help
Sources for: Synthesis/Implementation
Combinatoric_Example
xc3s200-4ft256
combExample - combinatoric_Struct (CombExam
inst1 - or3Gate - or3Arch (or3Gate.vhd)
inst2 - and2Gate - and2Arch (and2Gate.vhd)
inst3 - and2Gate - and2Arch (and2Gate.vhd)
combexample.ucf (combexample.ucf)
Sources
Snapshots
Libraries
Processes:
Create New Source
View Design Summary
Design Utilities
User Constraints
Create Timing Constraints
Assign Package Pins
Create Area Constraints
Edit Constraints (Text)
Synthesize - XST
View Synthesis Report
View RTL Schematic
View Technology Schematic
Check Syntax
Generate Post-Synthesis Simulation Model
Implement Design
Translate
Map
Place & Route
Generate Programming File
Programming File Generation Report
Generate PROM, ACE, or JTAG File
Configure Device (iMPACT)
Update Bitstream with Processor Data
Processes
Hierarchy - MyTestbench.tbw
Design Summary
MyTestbench.tbw
CombExample.vhd
combexample.ucf
Console
Errors
Warnings
Tcl Console
Find in Files
Ln 26 Col 6
CAPS
NUM
SCRL
VHDL

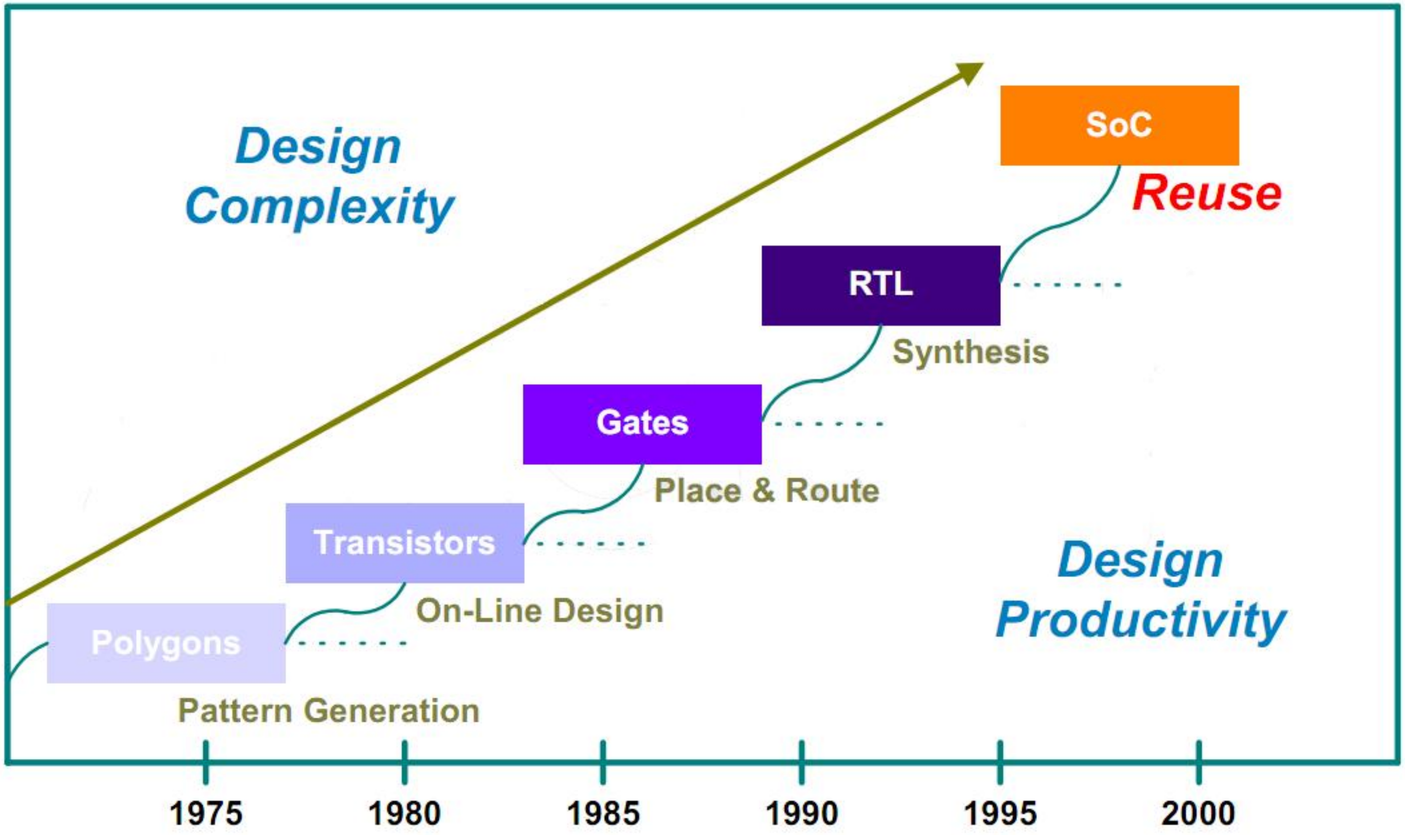
```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;


entity combExample is
   port (SW0, SW1, SW2, SW3, SW4: in std_logic;
         LDO: out std_logic);
end entity combExample;

architecture combinatoric_Struct of combExample is

   component and2Gate is
      port (A, B: in std_logic;
            C: out std_logic);
   end component and2Gate;

   component or3Gate is
      port (A, B, C: in std_logic;
            D: out std_logic);
   end component or3Gate;

   signal outOr, outAnd: std_logic;

begin

   inst1: or3Gate    port map (A => SW1, B => SW2, C => SW3, D => outOr);

   inst2: and2Gate   port map (A => SW0, B => SW4, C => outAnd);

   inst3: and2Gate   port map (A => outOr, B => outAnd, C => LDO);

end architecture combinatoric_Struct;


-- architecture Behavioral of combExample is
-- begin
-- LDO <= ((SW0 and SW4) and (SW1 or SW2 or SW3));
-- end Behavioral;
```

```
Started : "Generate Programming File".

Process "Generate Programming File" completed successfully
```

Evolution of FPGA

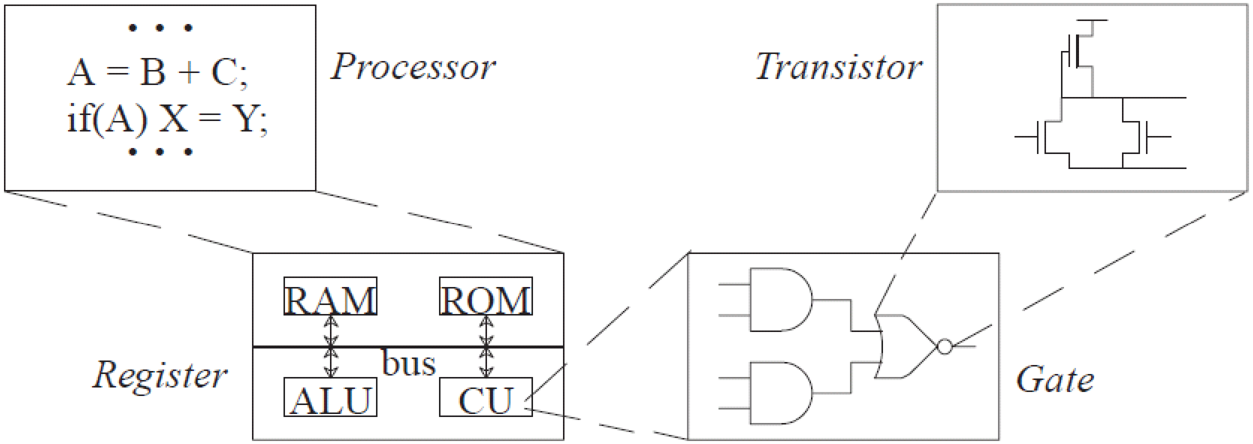


Trends in VLSI Design

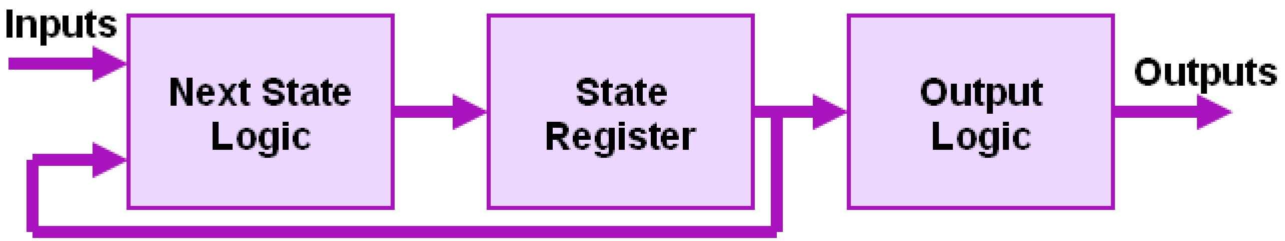


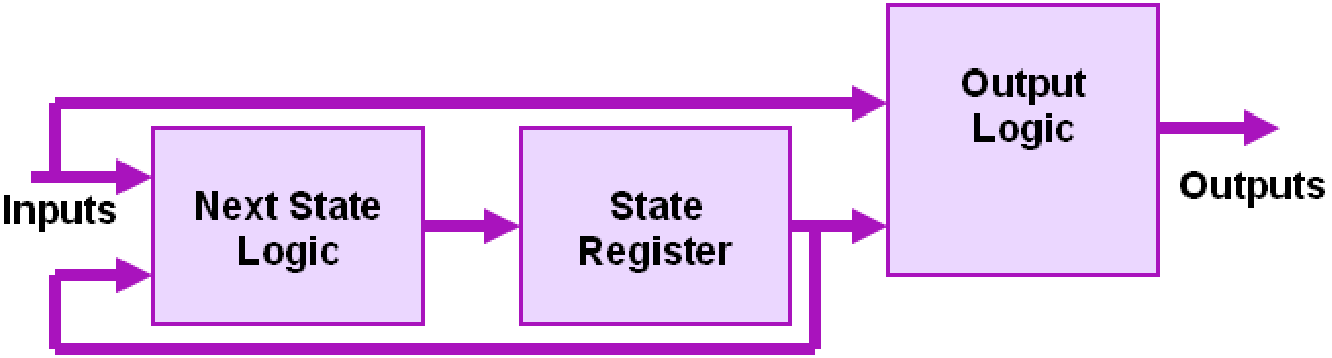
Levels of Abstraction

Levels	Behavioral forms	Structural components	Physical objects
Transistor	Differential eq., current–voltage diagrams	Transistors, resistors, capacitors	Analog & digital cells
Gate	Boolean equations, finite–state machines	Gates, flip–flops	Modules or units
Register	Algorithms, flowcharts, instruction sets, generalized FSM	Adders, comparators, registers, counters, register files, queues	Microchips
Processor	Executable specification, programs	Processors, controllers, memories, ASICs, ASIPs	Printed–circuit boards or multi–chip modules

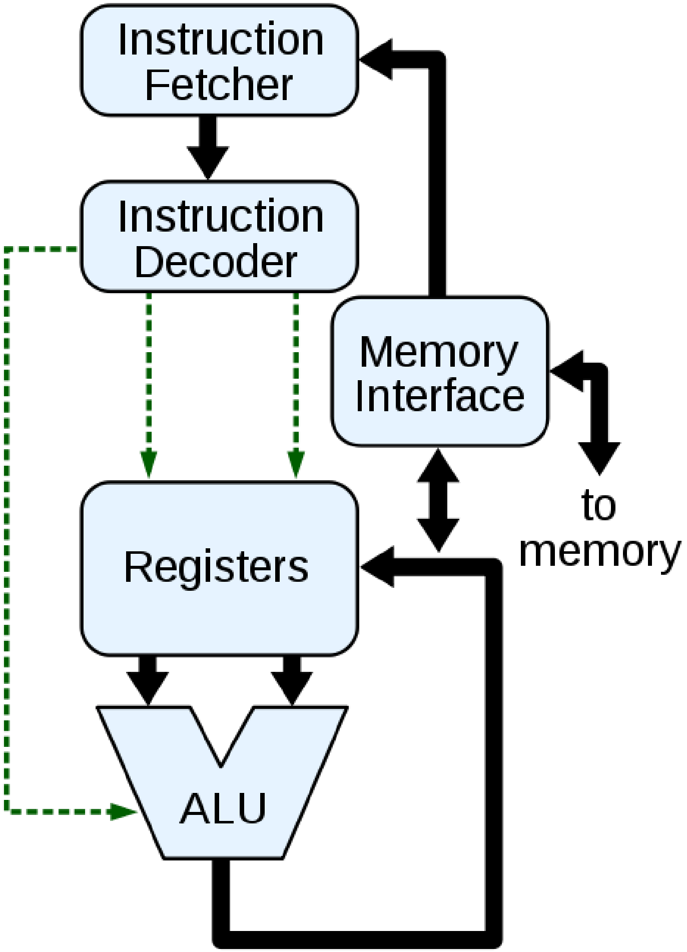


Finite State Machines

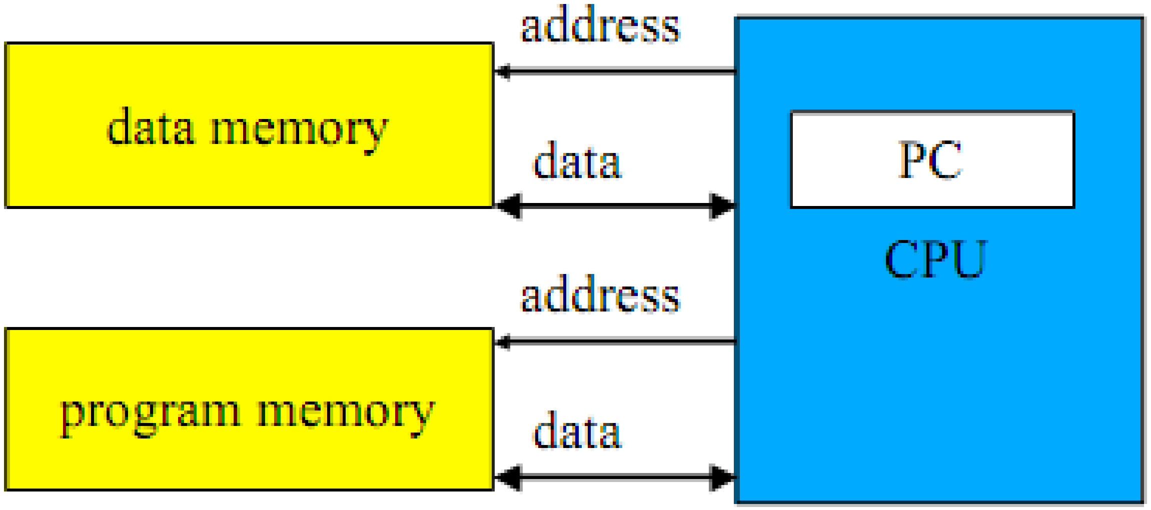




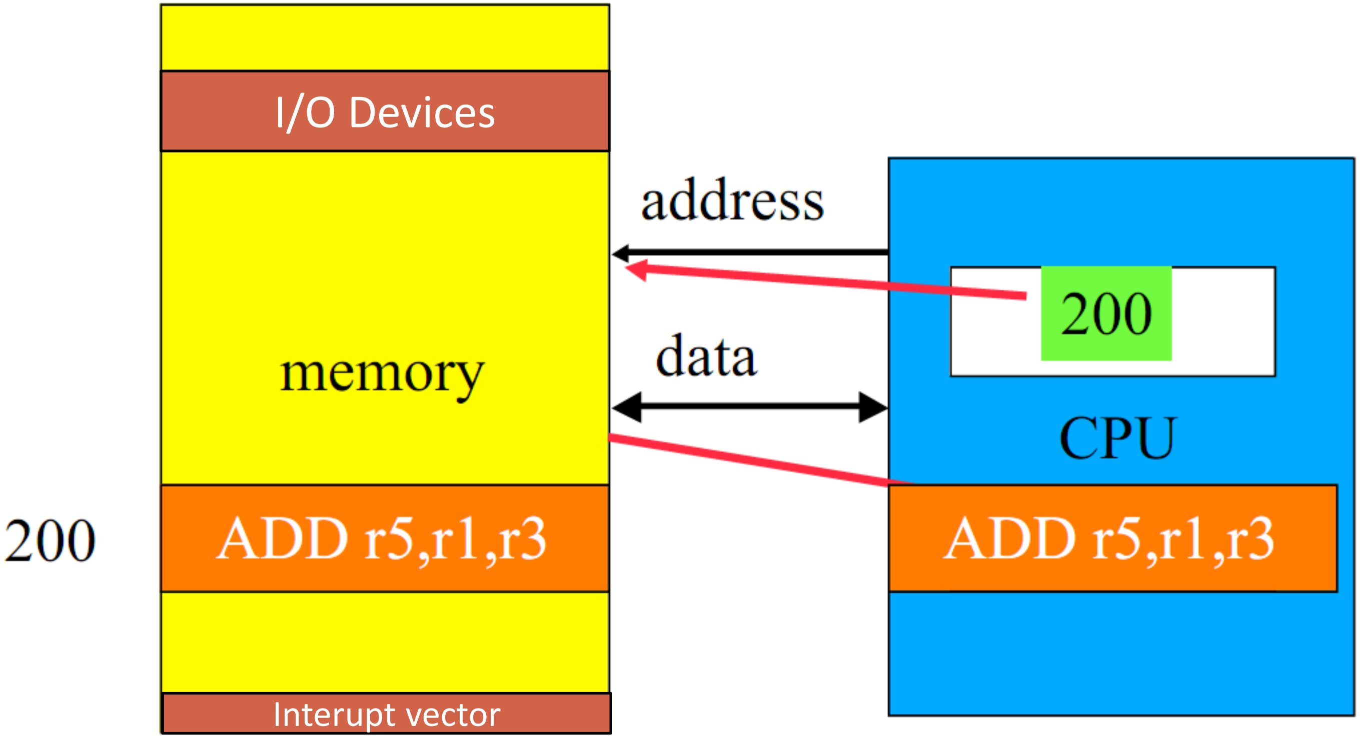
CPU – FSM Generalization



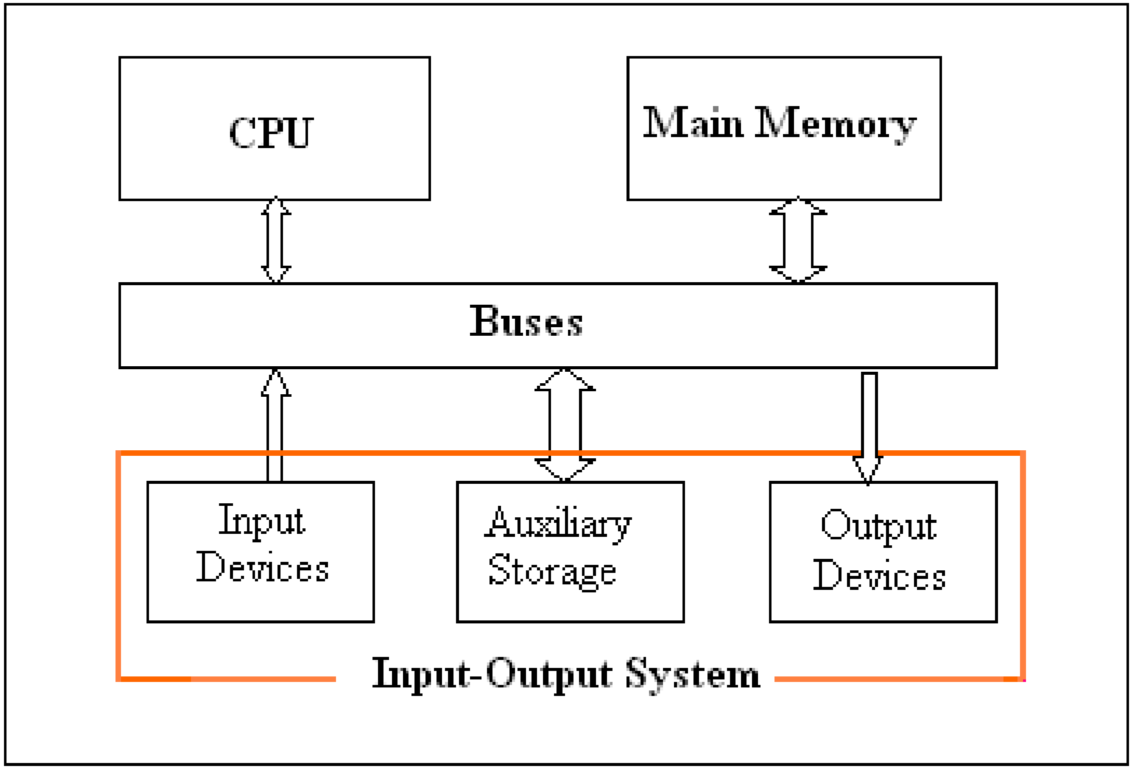
CPU – Harvard Architecture



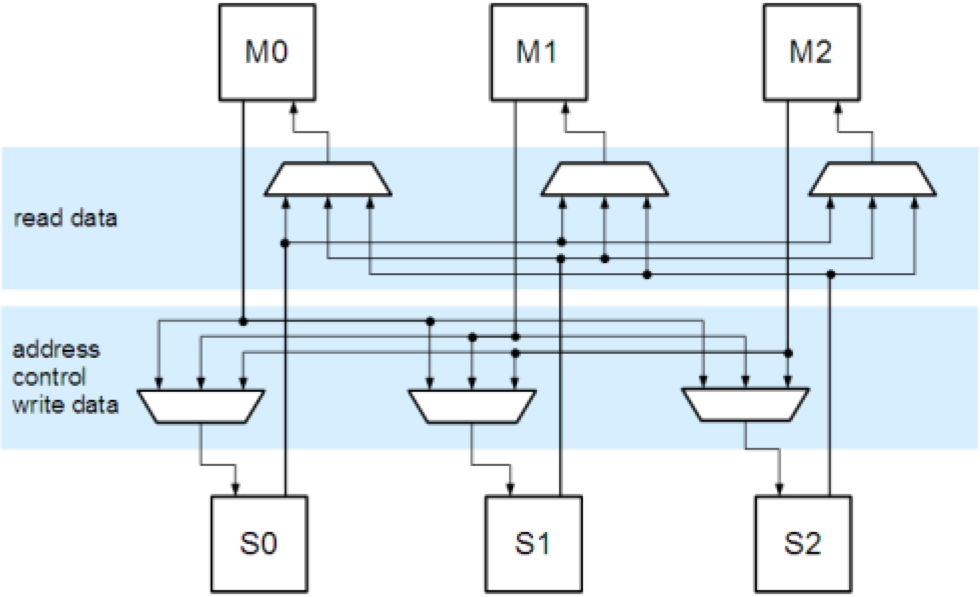
CPU – Memory

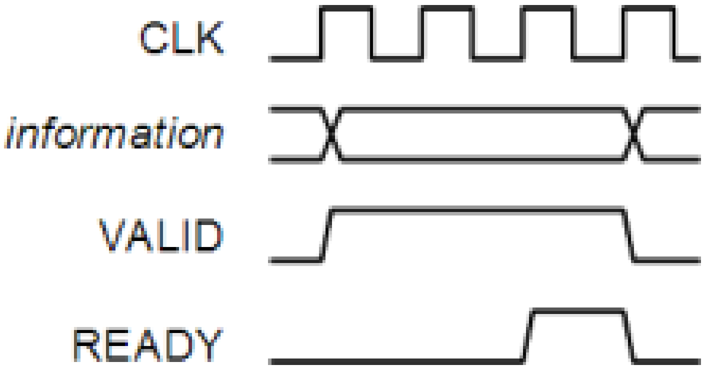

I/O Devices
address
data
200
memory
CPU
200
ADD r5,r1,r3
ADD r5,r1,r3
Interupt vector

A Bus

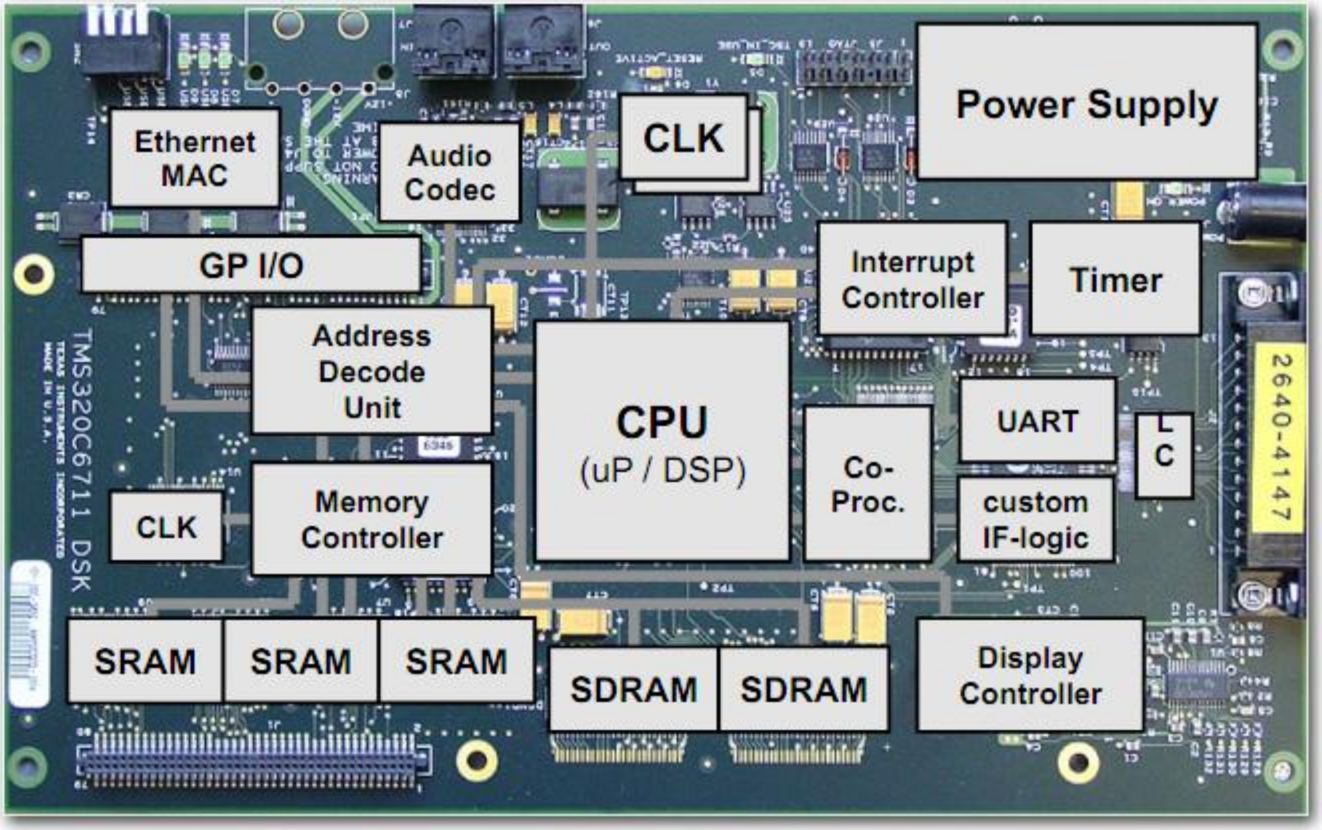

CPU
Main Memory
Buses
Input Devices
Auxiliary Storage
Output Devices
Input-Output System

AXI Bus


M0
M1
M2
read data
address
control
write data
S0
S1
S2

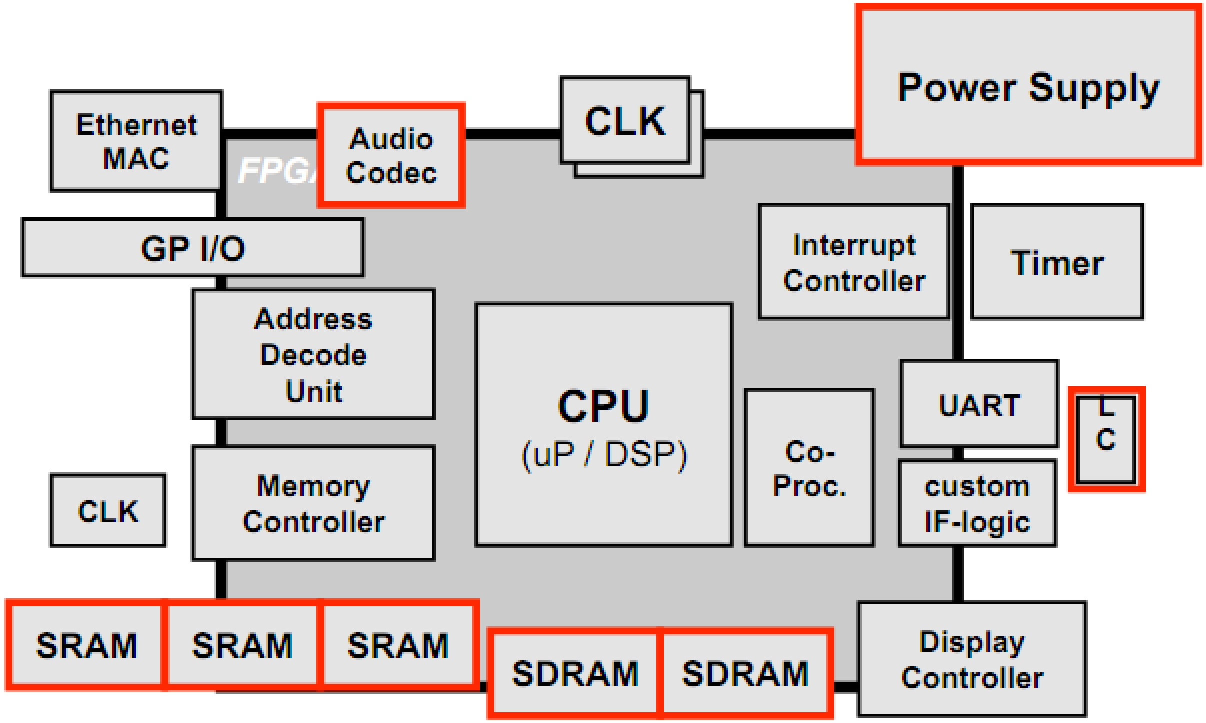

CLK
information
VALID
READY

Emended System

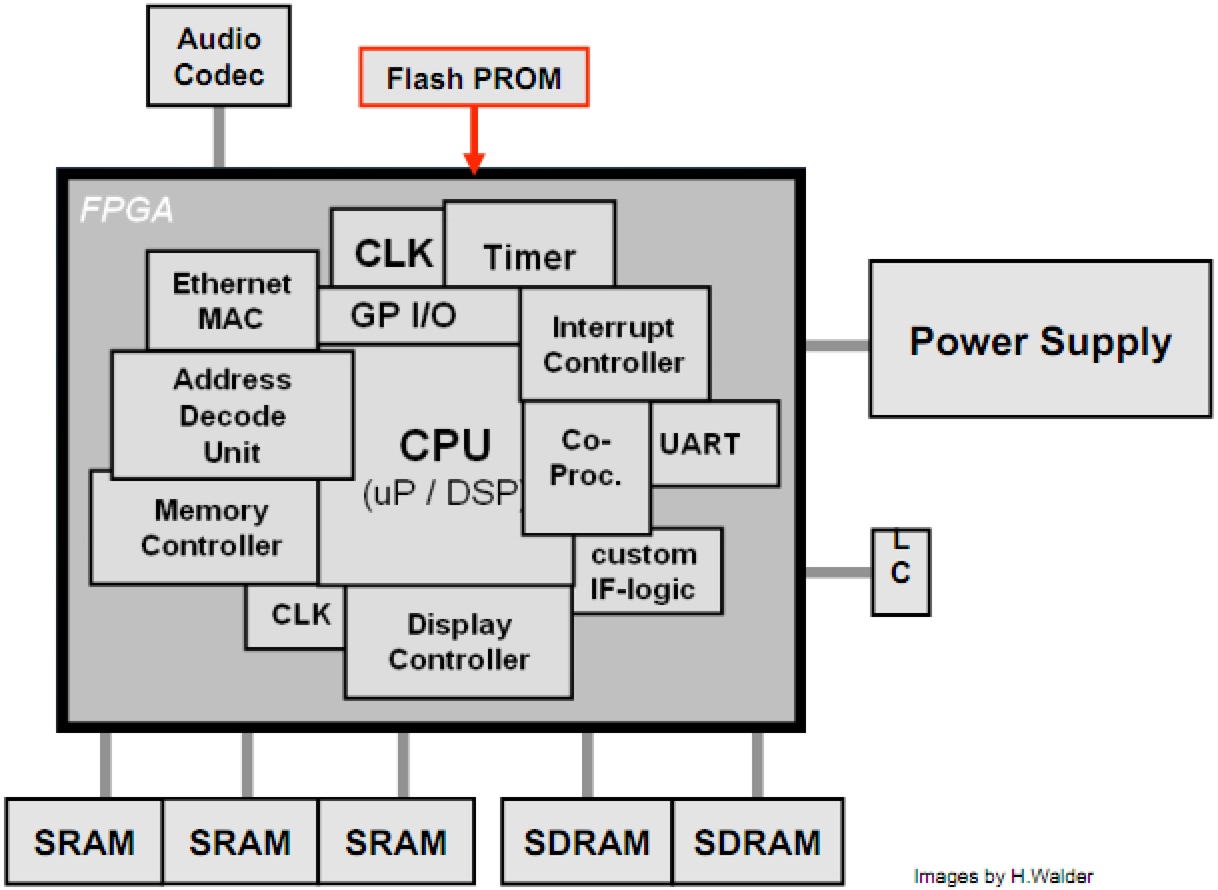


Images by H.Walder

Traditional Embedded System



Configurable System on Chip (CSoC)

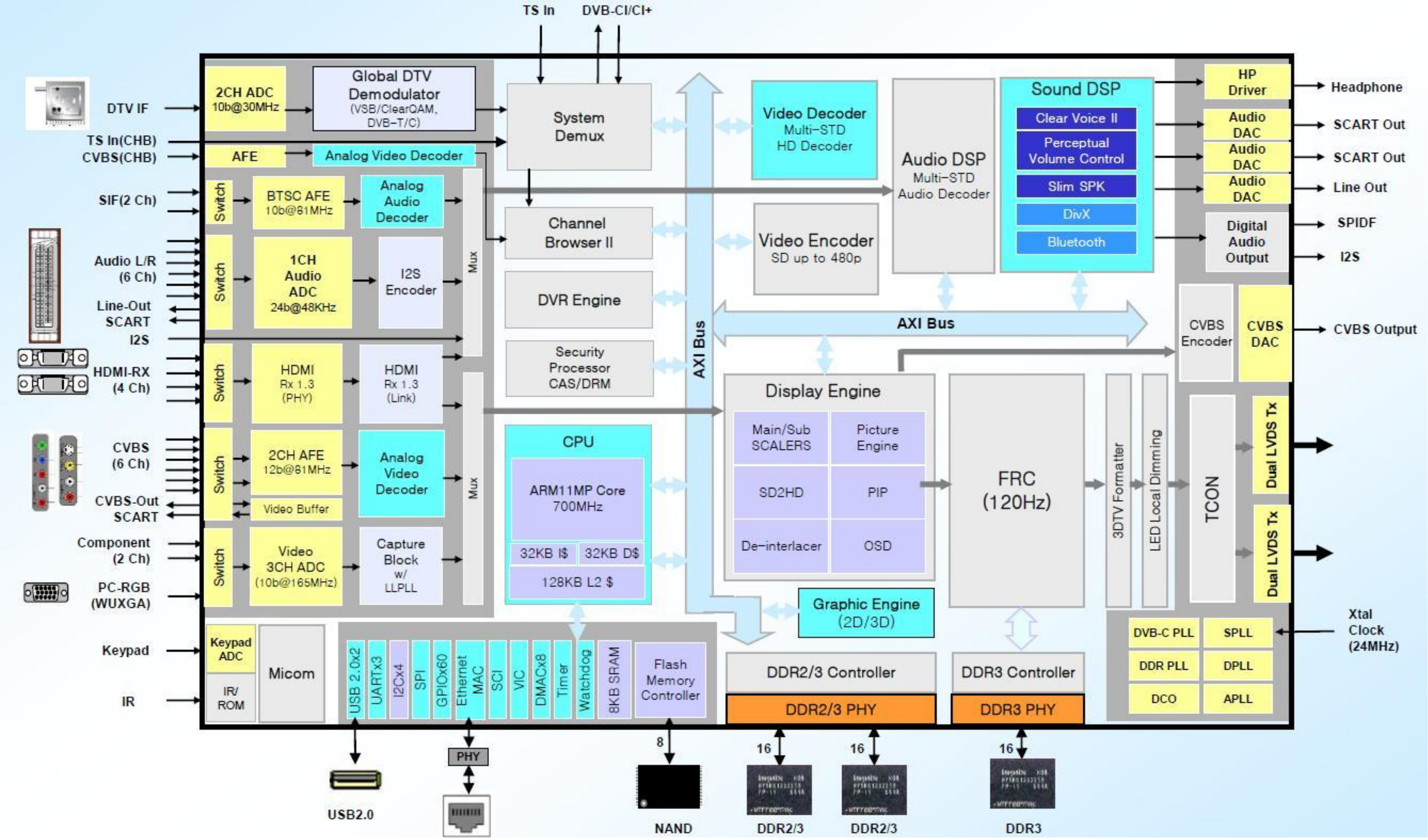


Images by H.Walder

Advantages

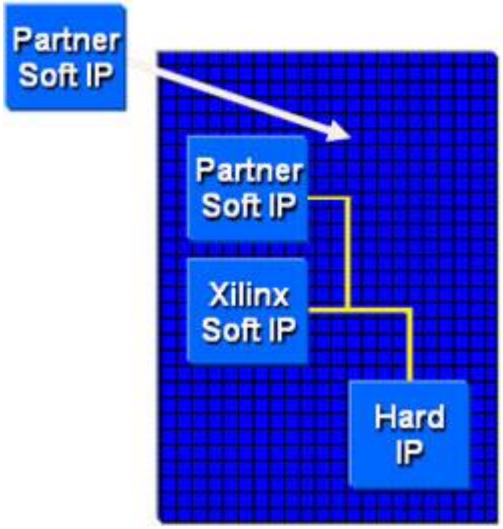
- Fewer physical components
- Shorter development cycles
- Field-programmable (updates, new features...)
- Custom co-processors / accelerators
- Possibly higher performance through on-chip integration
- Signals on a chip can typically be clocked higher than signals across board traces
- Optimization between modules possible
- Partial reconfigurability
- Exchange peripherals while the rest of the system keeps running

TV SoC example



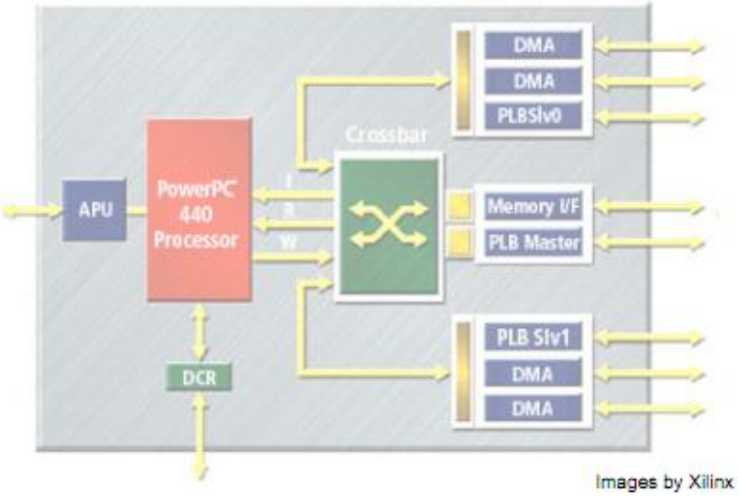
Building a SoC

- **HW-Blocks are called IP “cores”**
  - Buy them like Software
  - Implement own cores
  - Usually highly parametrizable
- **Interconnect infrastructure**
  - Bus Architectures
  - P2P Links
- **Cores implement (standardized) interfaces**
- **Modelling kit determines parameters for cores and “generates” the system**



Embedded CPUs

- **PowerPC 440 (hard core)**
  - 32 bit embedded PowerPC RISC
  - Up to 550 MHz
  - 2x 32 kB instruction and data caches
  - Memory management unit (MMU)
  - Hardware multiply and divide
  - Coprocessor interface (APU)
  - PLB and DCR bus interfaces

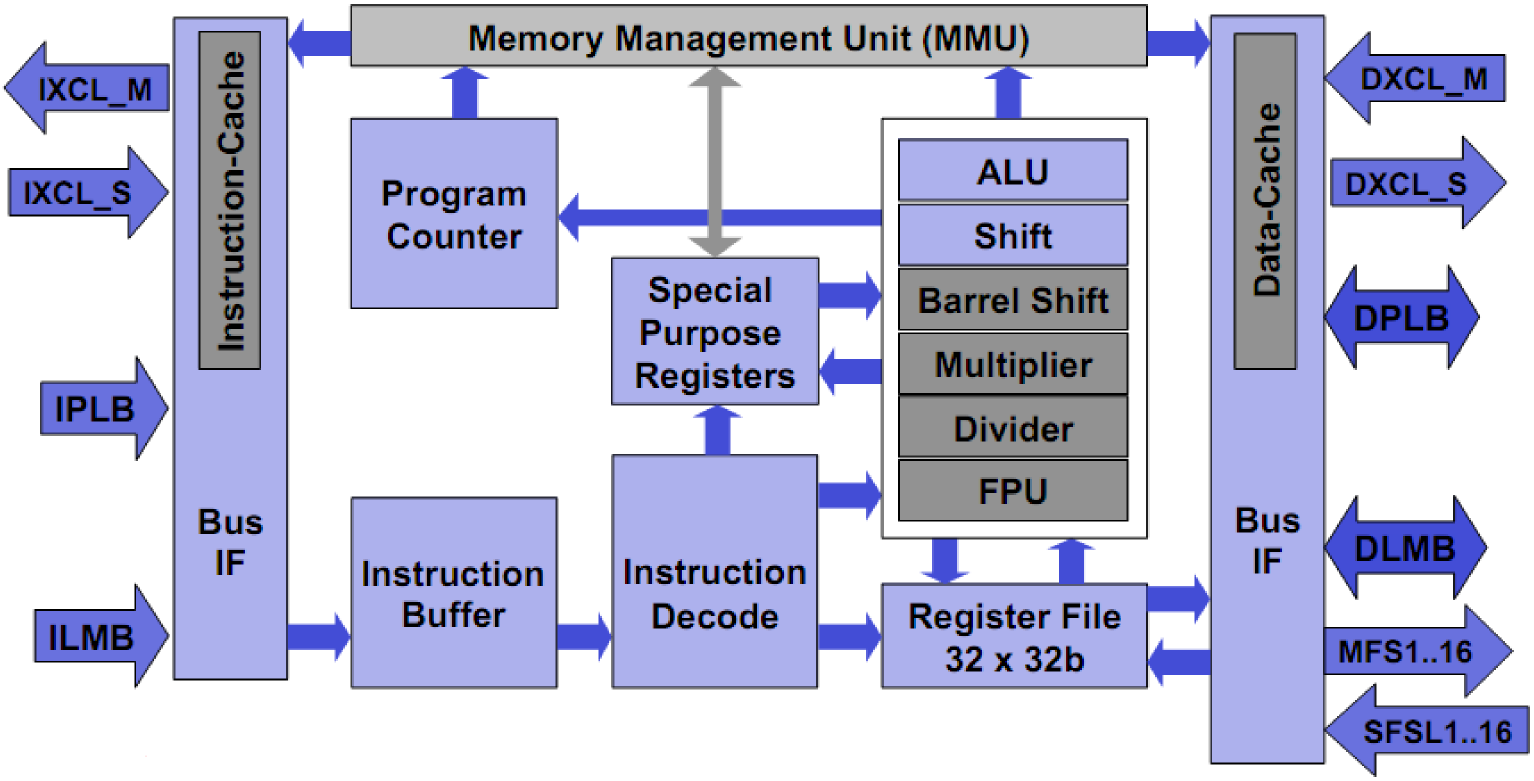


Images by Xilinx

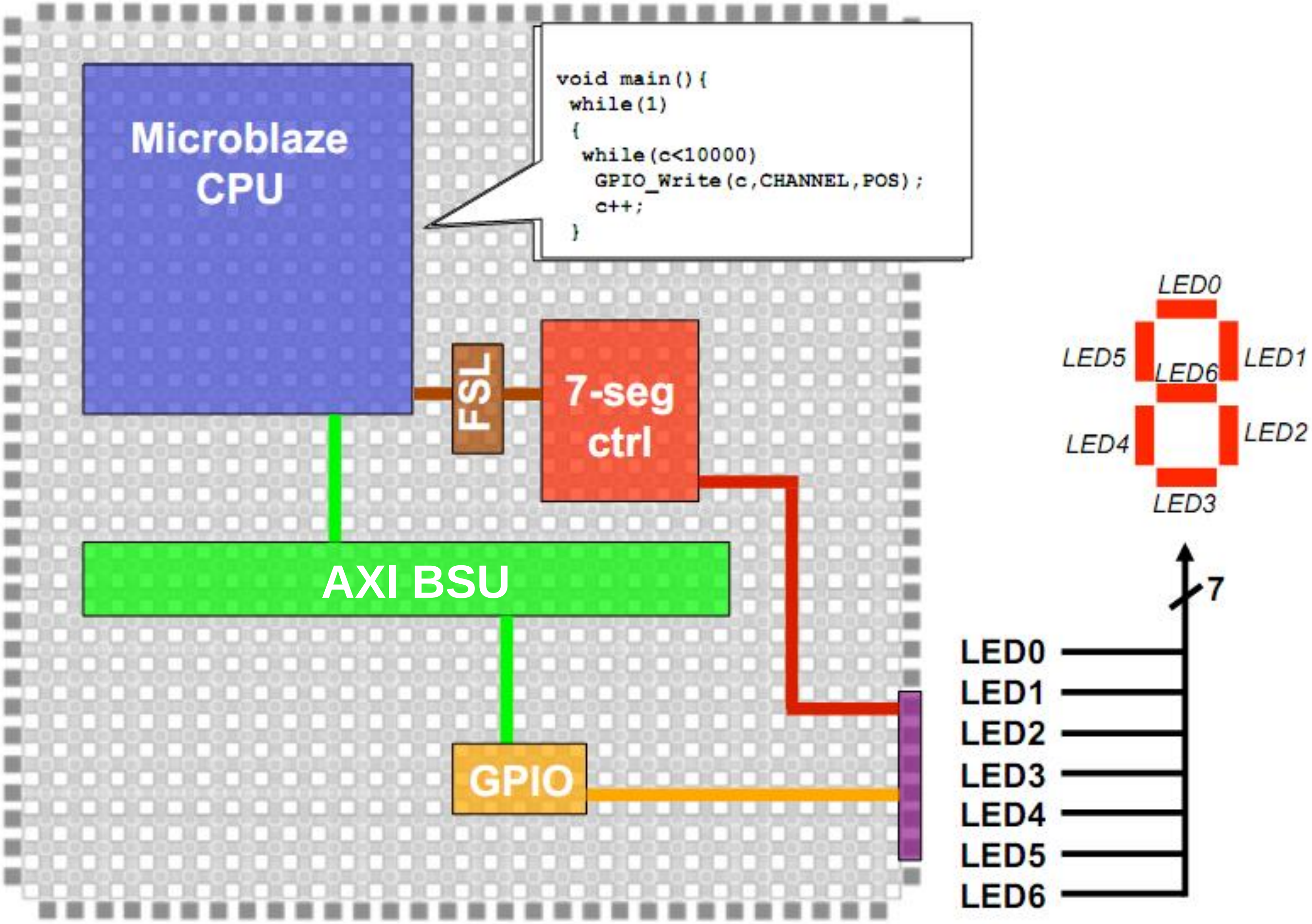
- **MicroBlaze (soft core)**
  - 32 bit RISC architecture, up to 2
  - Min. configuration: 1,010 LUTs [XC5VLX330 FPGA: 207,360 LU
  - Highly configurable
    - Barrel Shifter
    - MMU, 2-64 kB instruction and d
    - HW Multiplier/Divider, FPU, Deb
  - PLB, LMB, FSL bus interfaces

- **Others**
  - NIOS (Altera)
  - ARM
  - PicoBlaze (Xilinx), ...

Microblaze Configuration

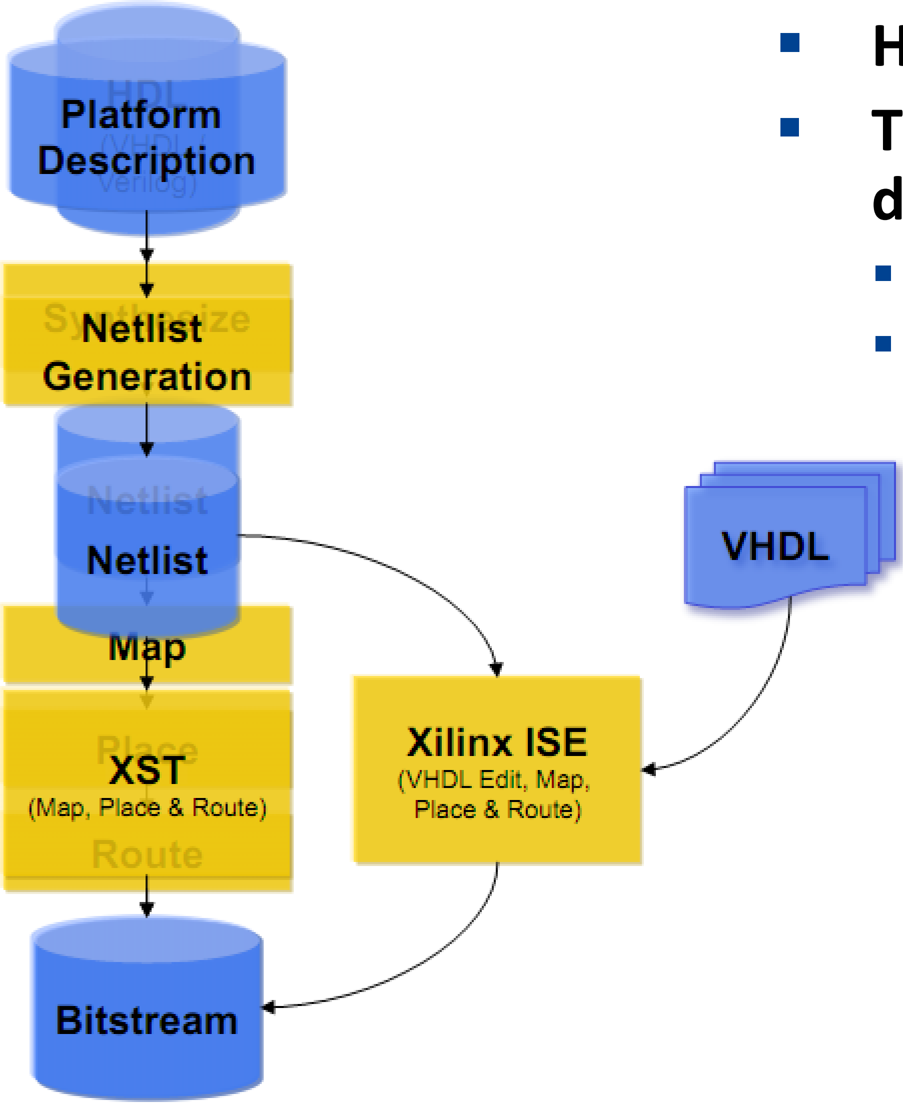


- High bandwith - Processor Bus (PLB/AXI)
- Configurable caches - Xilinx Cache Link (XCL)
- Local (BRAM) memory - Local Memory Bus (LMB)
- Co-processor port - Fast Simplex Link (FSL)



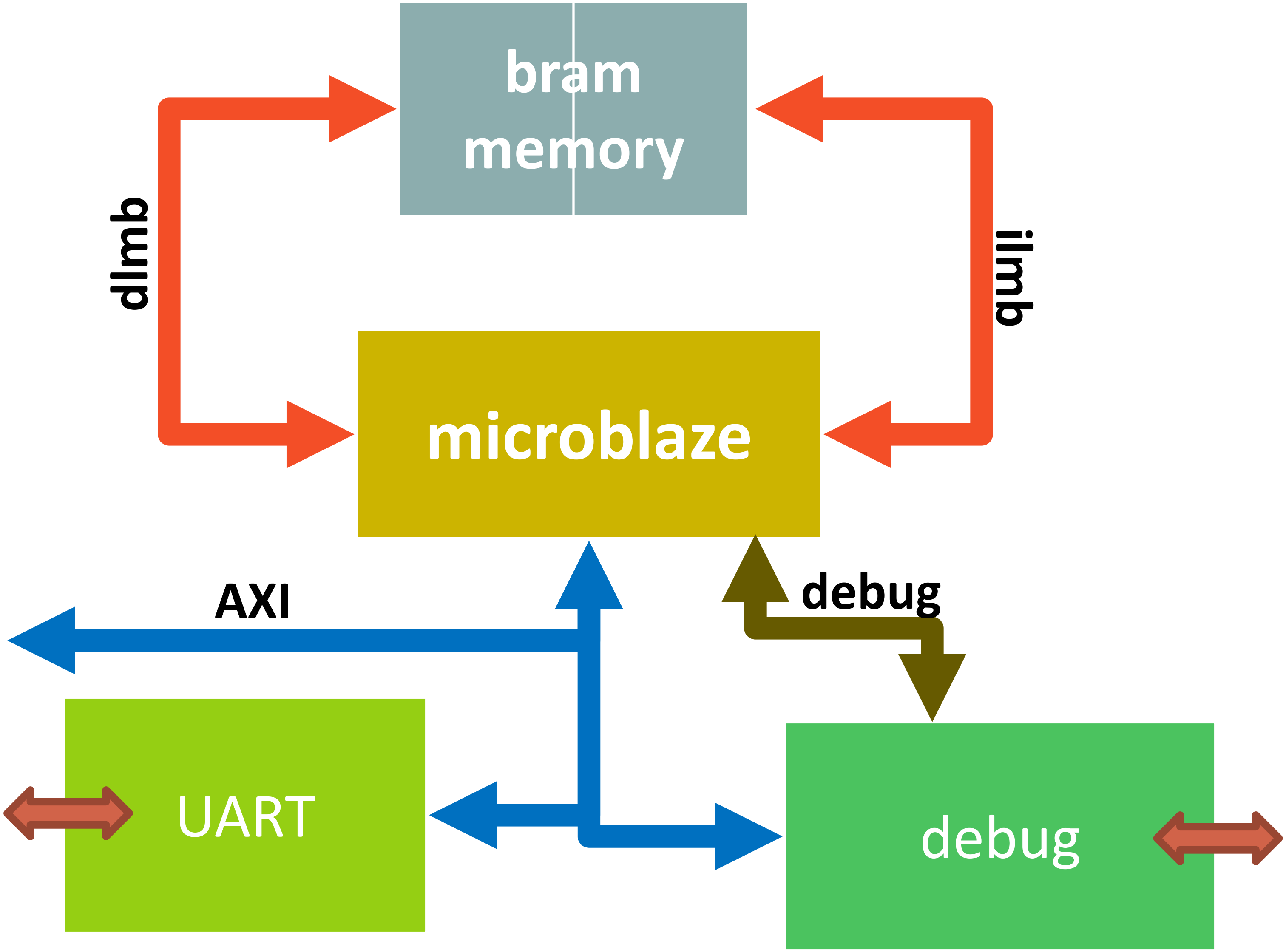
```
void main(){
 while(1)
 {
  while(c<10000)
   GPIO_Write(c,CHANNEL,POS);
   c++;
 }
```

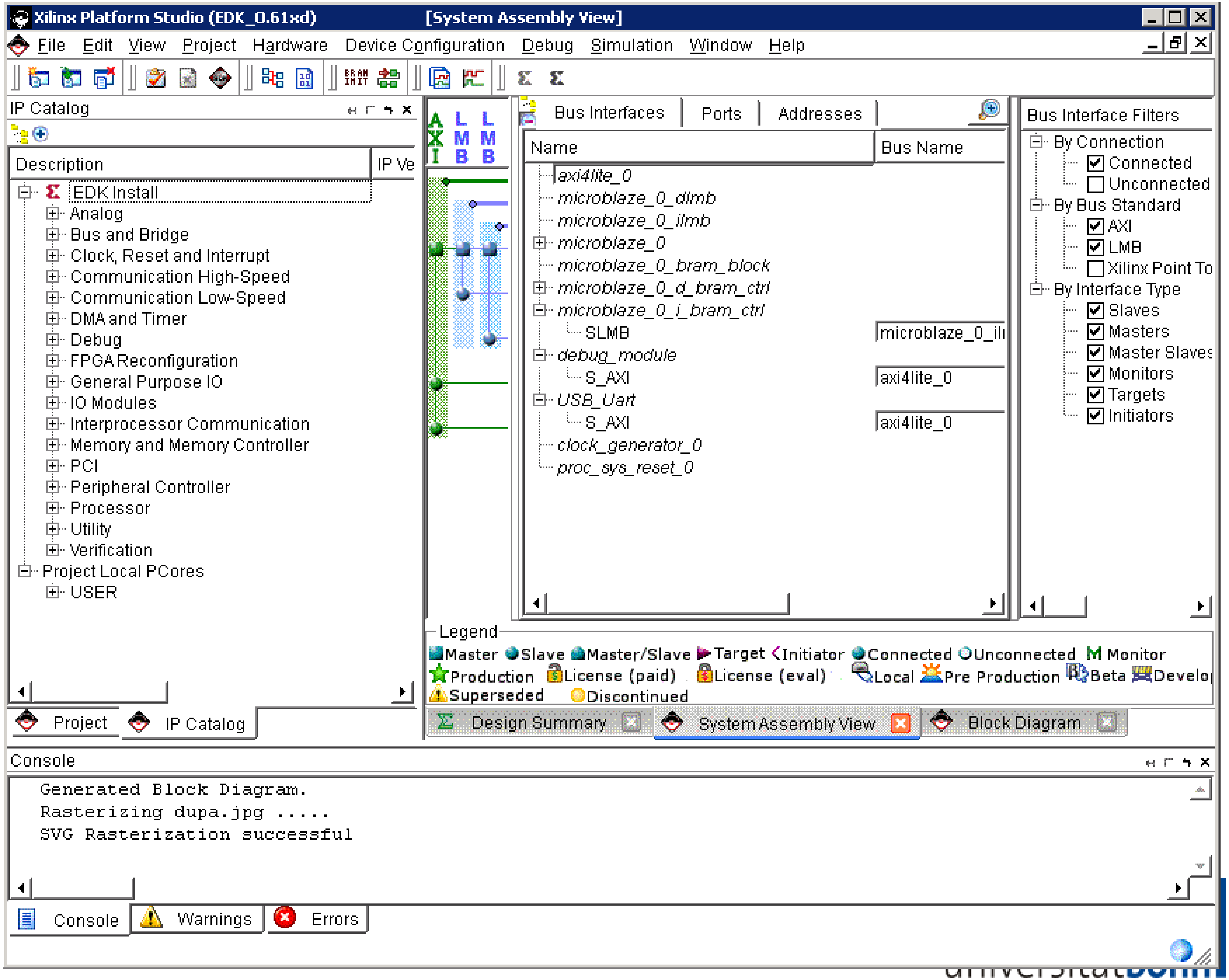
CSoC Hardware Design Flow



- **Higher level of abstraction**
- **Transform platform description into netlist**
  - Map, place & route to FPGA, or
  - Import in ISE and used in a larger FPGA design

EDK – our system





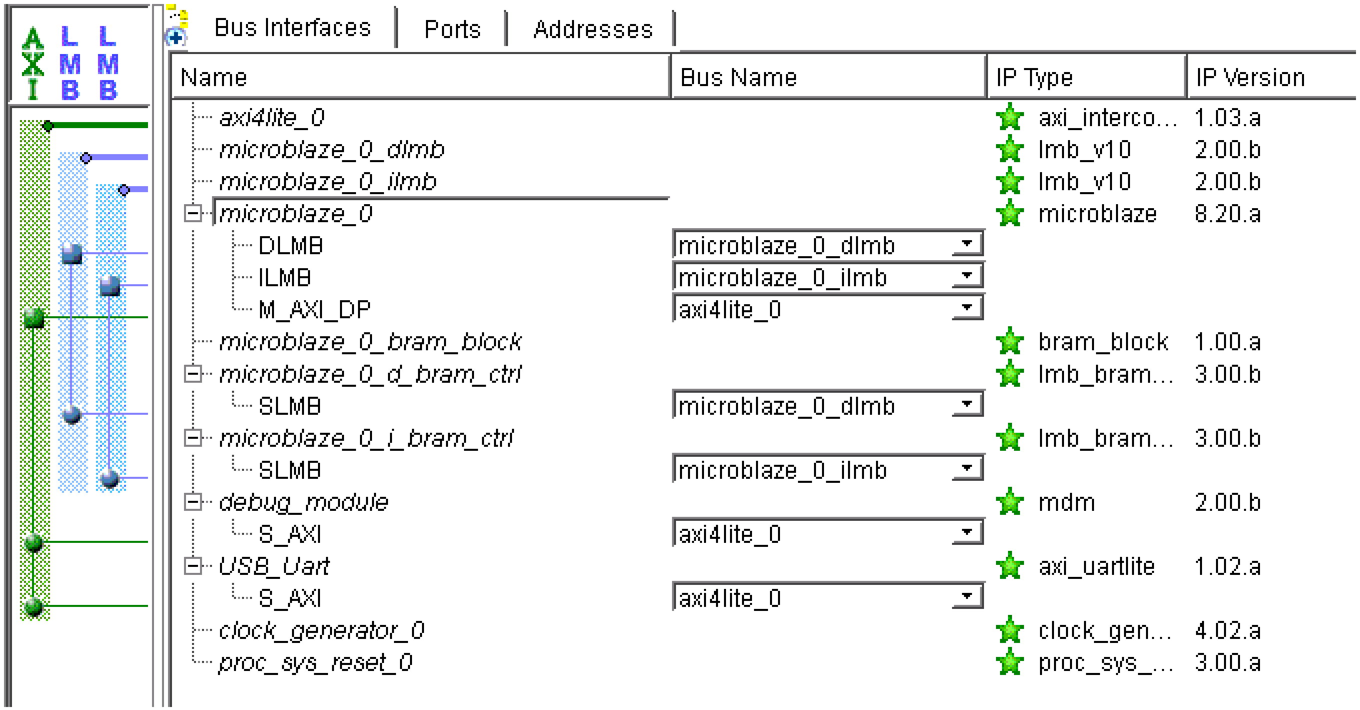
Name	Bus Name
axi4lite_0	
microblaze_0_dlmb	
microblaze_0_ilmb	
microblaze_0	
microblaze_0_bram_block	
microblaze_0_d_bram_ctrl	
microblaze_0_i_bram_ctrl	
SLMB	microblaze_0_ili
debug_module	
S_AXI	axi4lite_0
USB_Uart	
S_AXI	axi4lite_0
clock_generator_0	
proc_sys_reset_0	

Bus Interface Filters

- By Connection
  - ☑ Connected
  - □ Unconnected
- By Bus Standard
  - ☑ AXI
  - ☑ LMB
  - □ Xilinx Point To
- By Interface Type
  - ☑ Slaves
  - ☑ Masters
  - ☑ Master Slaves
  - ☑ Monitors
  - ☑ Targets
  - ☑ Initiators

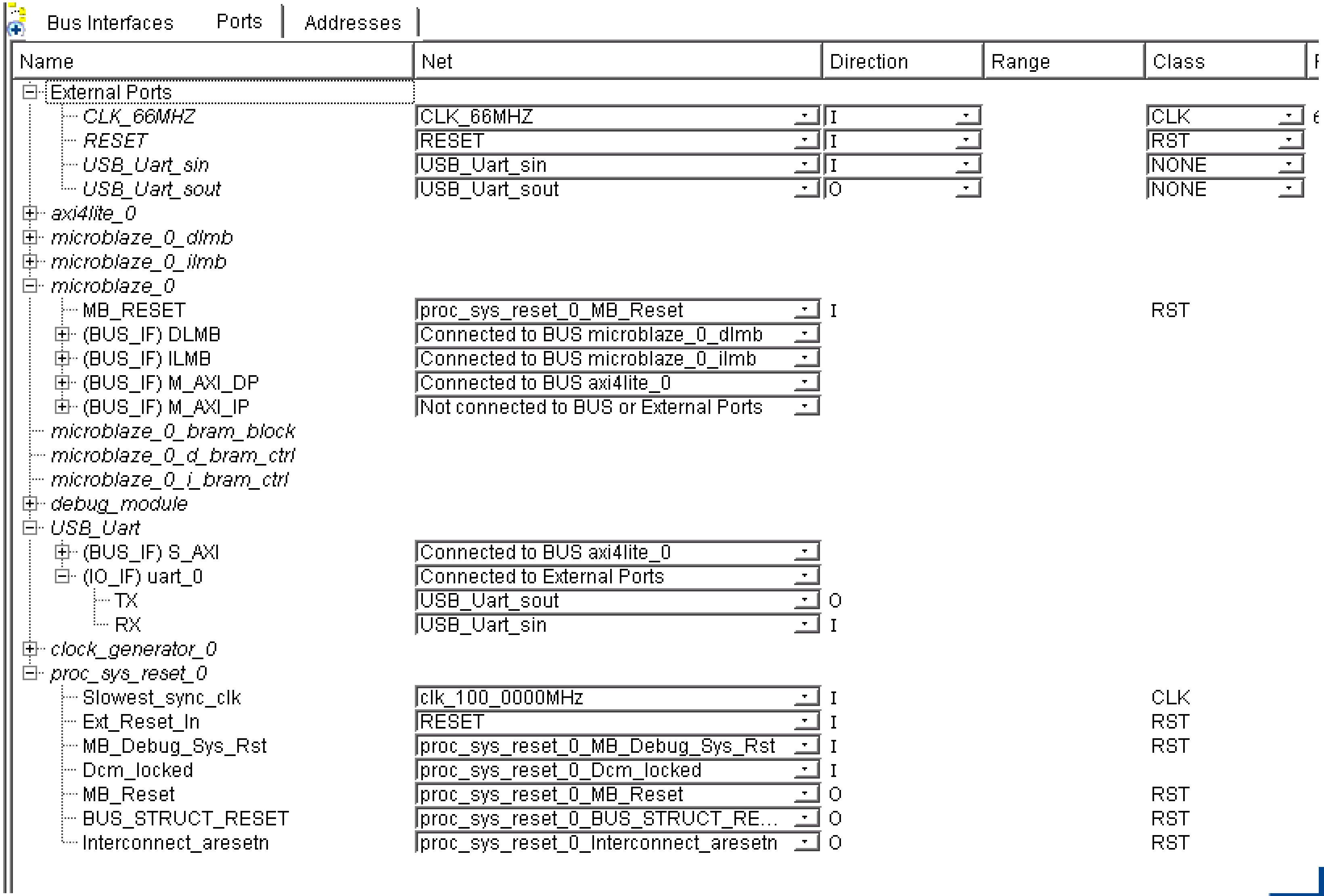
```
Generated Block Diagram.
Rasterizing dupa.jpg .....
SVG Rasterization successful
```

EDK – Bus Interface



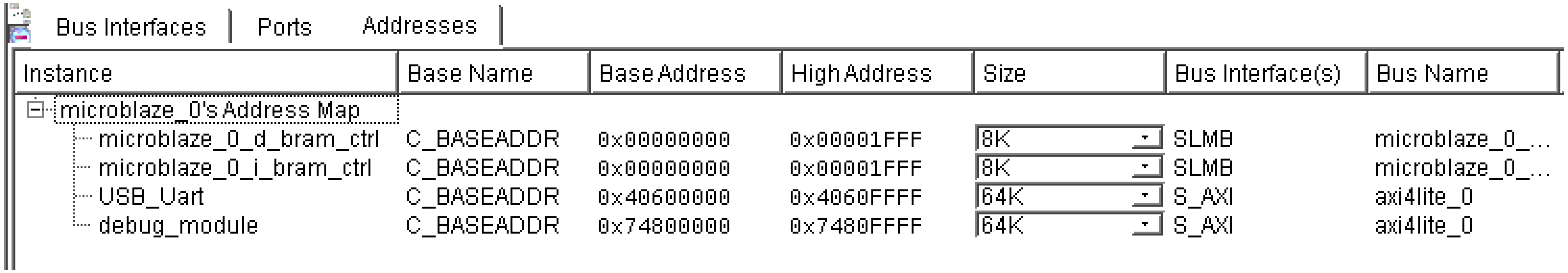
Name	Bus Name	IP Type	IP Version
axi4lite_0		axi_interco...	1.03.a
microblaze_0_dlmb		lmb_v10	2.00.b
microblaze_0_ilmb		lmb_v10	2.00.b
microblaze_0		microblaze	8.20.a
DLMB	microblaze_0_dlmb		
ILMB	microblaze_0_ilmb		
M_AXI_DP	axi4lite_0		
microblaze_0_bram_block		bram_block	1.00.a
microblaze_0_d_bram_ctrl		lmb_bram...	3.00.b
SLMB	microblaze_0_dlmb		
microblaze_0_i_bram_ctrl		lmb_bram...	3.00.b
SLMB	microblaze_0_ilmb		
debug_module		mdm	2.00.b
S_AXI	axi4lite_0		
USB_Uart		axi_uartlite	1.02.a
S_AXI	axi4lite_0		
clock_generator_0		clock_gen...	4.02.a
proc_sys_reset_0		proc_sys_...	3.00.a

EDK - Ports



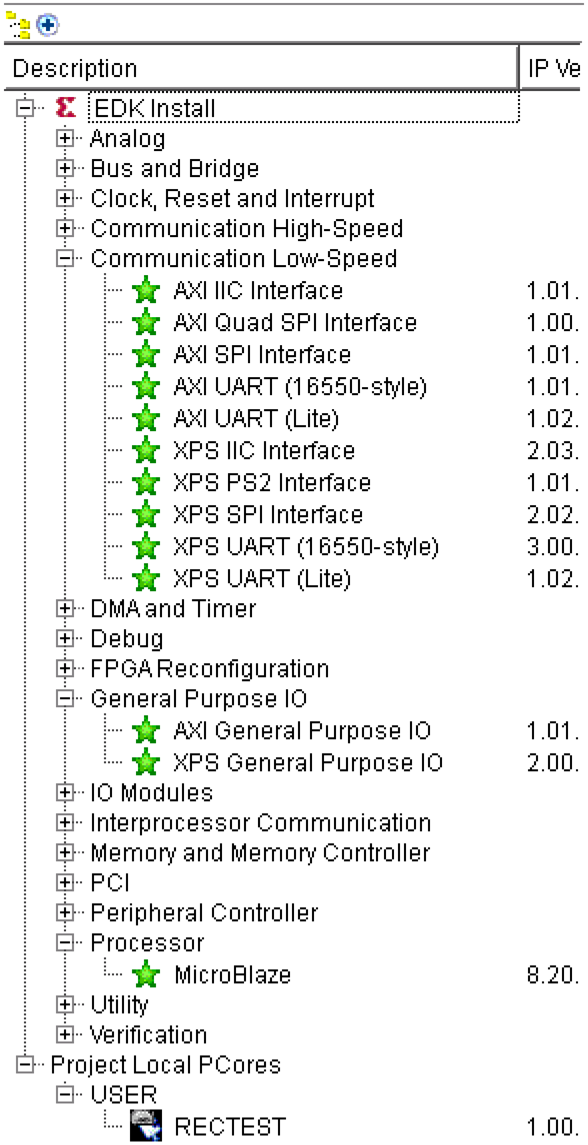
Name	Net	Direction	Range	Class	F
⊟ External Ports					
CLK_66MHZ	CLK_66MHZ	I		CLK	6
RESET	RESET	I		RST	
USB_Uart_sin	USB_Uart_sin	I		NONE	
USB_Uart_sout	USB_Uart_sout	O		NONE	
⊞ *axi4lite_0*					
⊞ *microblaze_0_dlmb*					
⊞ *microblaze_0_ilmb*					
⊟ *microblaze_0*					
MB_RESET	proc_sys_reset_0_MB_Reset	I		RST	
⊞ (BUS_IF) DLMB	Connected to BUS microblaze_0_dlmb				
⊞ (BUS_IF) ILMB	Connected to BUS microblaze_0_ilmb				
⊞ (BUS_IF) M_AXI_DP	Connected to BUS axi4lite_0				
⊞ (BUS_IF) M_AXI_IP	Not connected to BUS or External Ports				
microblaze_0_bram_block					
microblaze_0_d_bram_ctrl					
microblaze_0_i_bram_ctrl					
⊞ *debug_module*					
⊟ *USB_Uart*					
⊞ (BUS_IF) S_AXI	Connected to BUS axi4lite_0				
⊟ (IO_IF) uart_0	Connected to External Ports				
TX	USB_Uart_sout	O			
RX	USB_Uart_sin	I			
⊞ *clock_generator_0*					
⊟ *proc_sys_reset_0*					
Slowest_sync_clk	clk_100_0000MHz	I		CLK	
Ext_Reset_In	RESET	I		RST	
MB_Debug_Sys_Rst	proc_sys_reset_0_MB_Debug_Sys_Rst	I		RST	
Dcm_locked	proc_sys_reset_0_Dcm_locked	I			
MB_Reset	proc_sys_reset_0_MB_Reset	O		RST	
BUS_STRUCT_RESET	proc_sys_reset_0_BUS_STRUCT_RE...	O		RST	
Interconnect_aresetn	proc_sys_reset_0_Interconnect_aresetn	O		RST	

EDK - Adresses



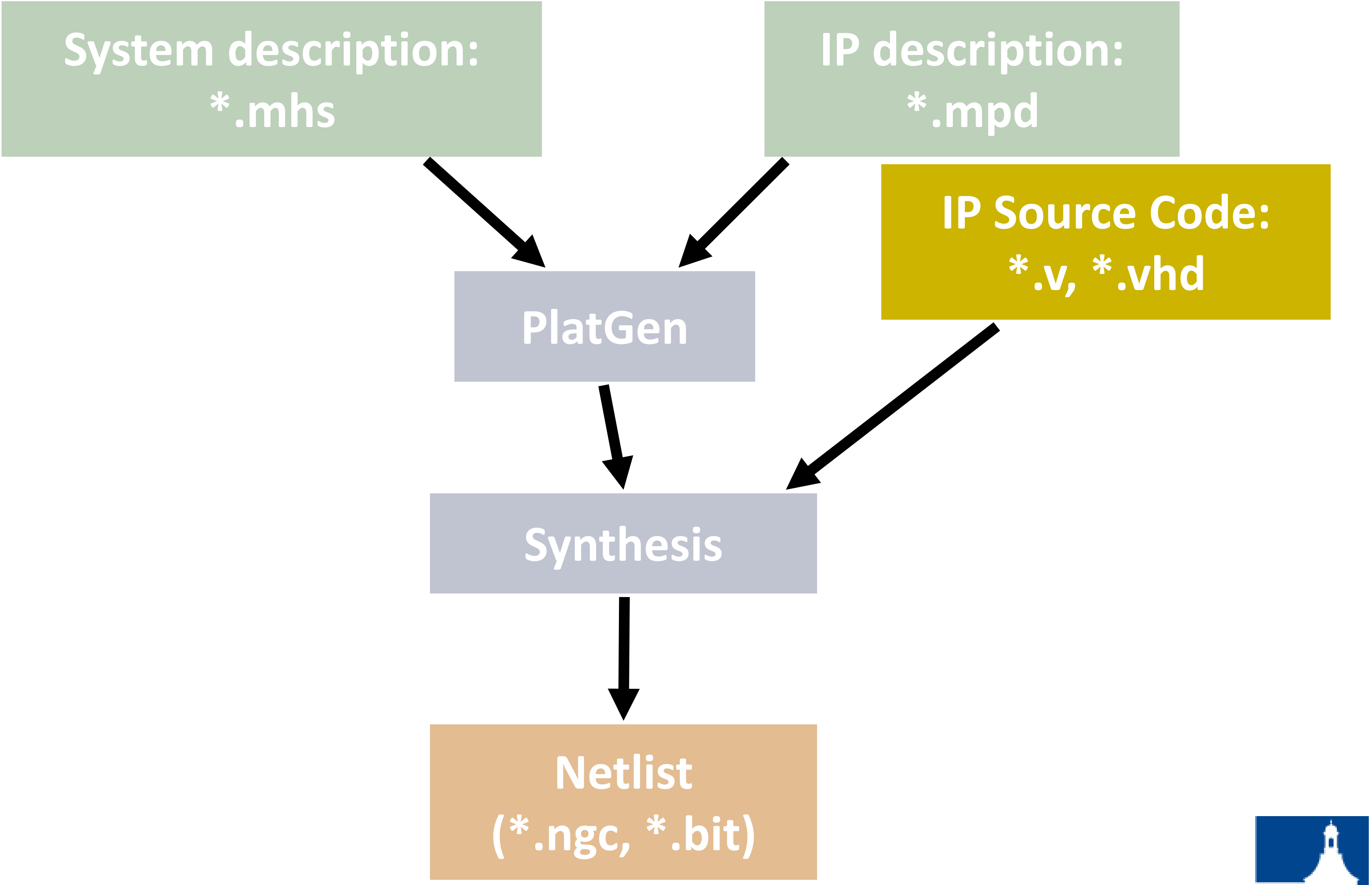
Instance	Base Name	Base Address	High Address	Size	Bus Interface(s)	Bus Name
⊟ microblaze_0's Address Map						
microblaze_0_d_bram_ctrl	C_BASEADDR	0x00000000	0x00001FFF	8K	SLMB	microblaze_0_...
microblaze_0_i_bram_ctrl	C_BASEADDR	0x00000000	0x00001FFF	8K	SLMB	microblaze_0_...
USB_Uart	C_BASEADDR	0x40600000	0x4060FFFF	64K	S_AXI	axi4lite_0
debug_module	C_BASEADDR	0x74800000	0x7480FFFF	64K	S_AXI	axi4lite_0

EDK – IP Catalog



```
Description                                   IP Ve
⊟ EDK Install
   ⊞ Analog
   ⊞ Bus and Bridge
   ⊞ Clock, Reset and Interrupt
   ⊞ Communication High-Speed
   ⊟ Communication Low-Speed
        AXI IIC Interface                     1.01.
        AXI Quad SPI Interface                1.00.
        AXI SPI Interface                     1.01.
        AXI UART (16550-style)                1.01.
        AXI UART (Lite)                       1.02.
        XPS IIC Interface                     2.03.
        XPS PS2 Interface                     1.01.
        XPS SPI Interface                     2.02.
        XPS UART (16550-style)                3.00.
        XPS UART (Lite)                       1.02.
   ⊞ DMA and Timer
   ⊞ Debug
   ⊞ FPGA Reconfiguration
   ⊟ General Purpose IO
        AXI General Purpose IO                1.01.
        XPS General Purpose IO                2.00.
   ⊞ IO Modules
   ⊞ Interprocessor Communication
   ⊞ Memory and Memory Controller
   ⊞ PCI
   ⊞ Peripheral Controller
   ⊟ Processor
        MicroBlaze                            8.20.
   ⊞ Utility
   ⊞ Verification
⊟ Project Local PCores
   ⊟ USER
        RECTEST                               1.00.
```

Backend Files



System description:
*.mhs

IP description:
*.mpd

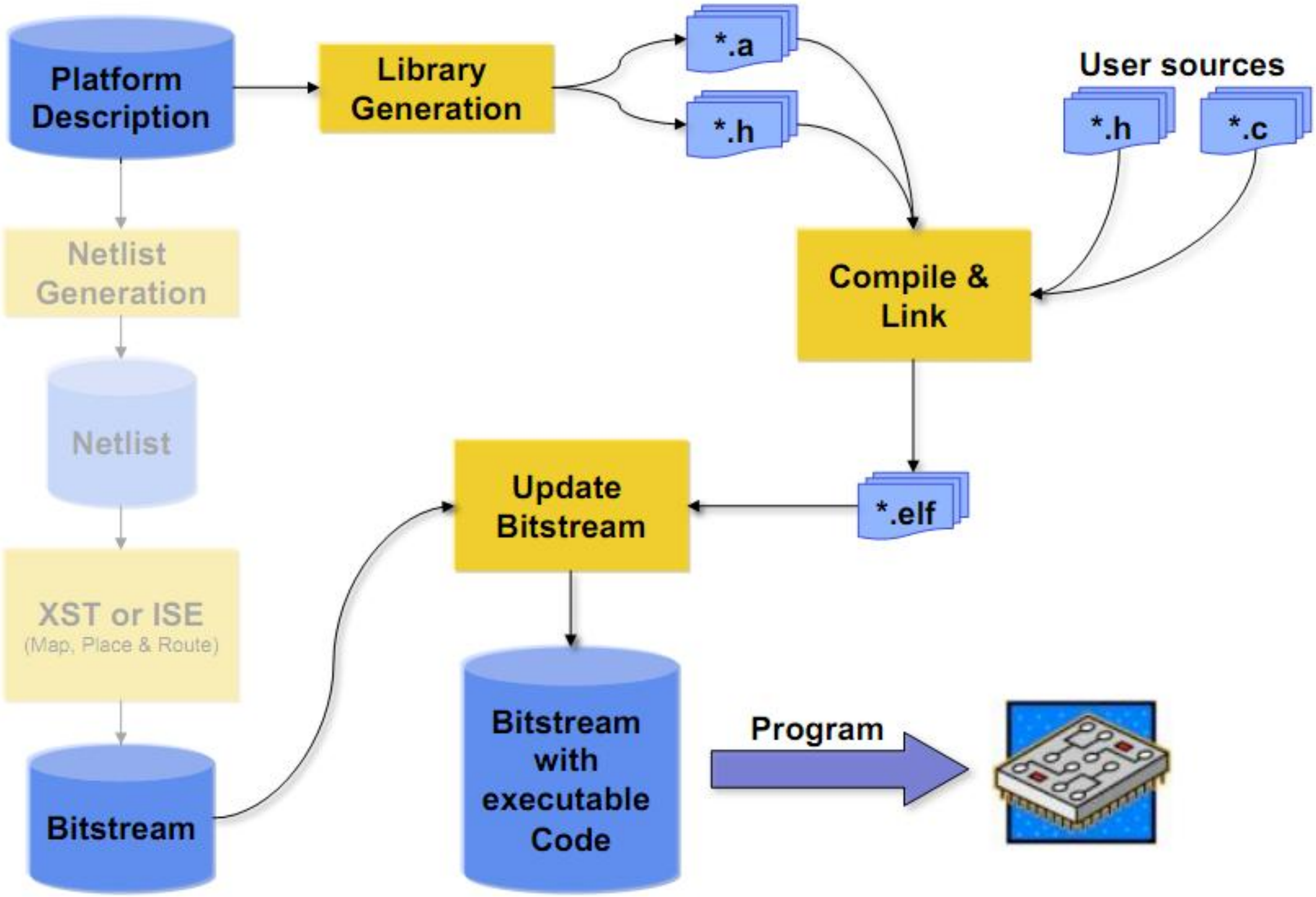
IP Source Code:
*.v, *.vhd

PlatGen

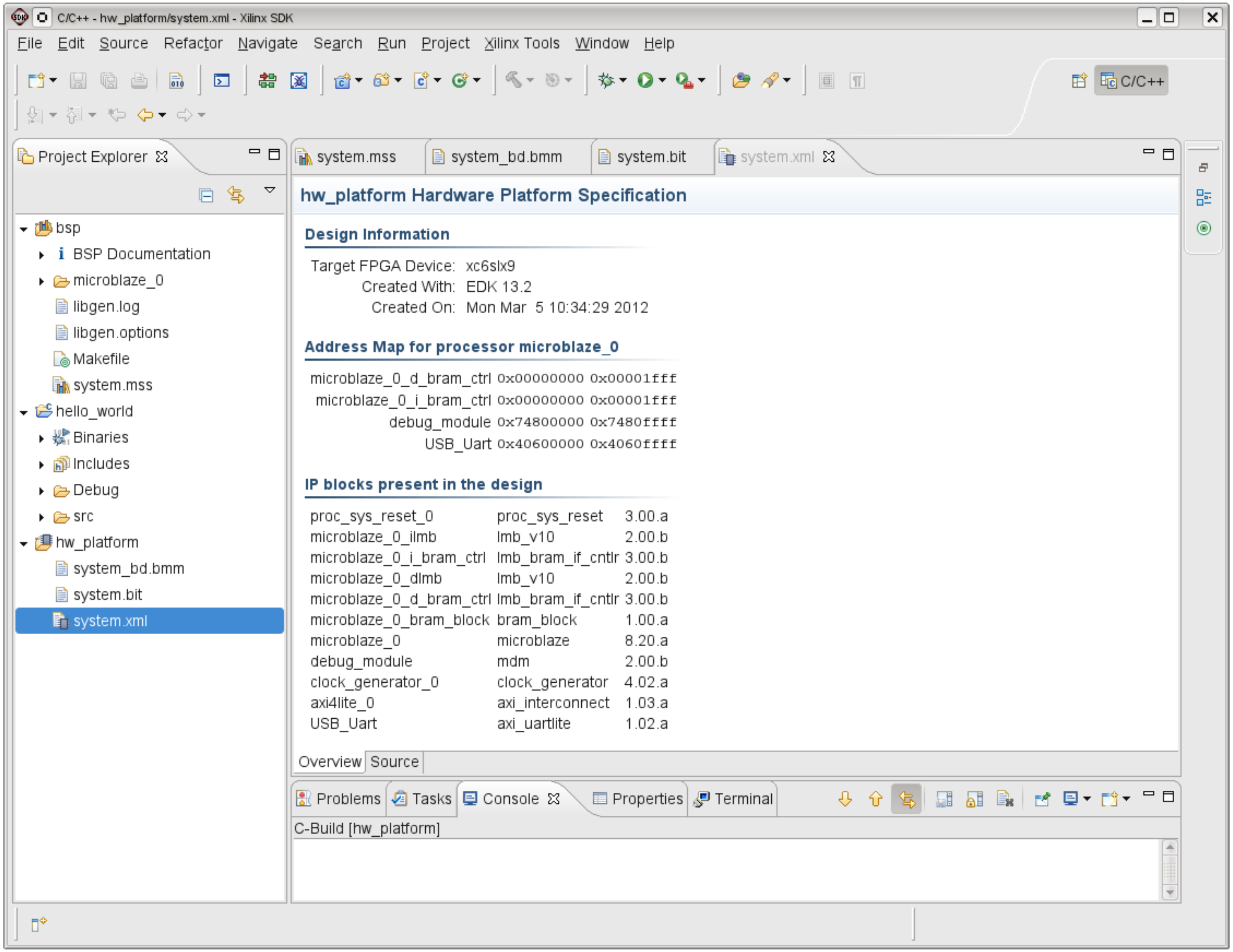
Synthesis

Netlist
(*.ngc, *.bit)

CSoC Software Design Flow



xSDK – HW platform



hw_platform Hardware Platform Specification

Design Information

```
Target FPGA Device:  xc6slx9
      Created With:  EDK 13.2
        Created On:  Mon Mar  5 10:34:29 2012
```

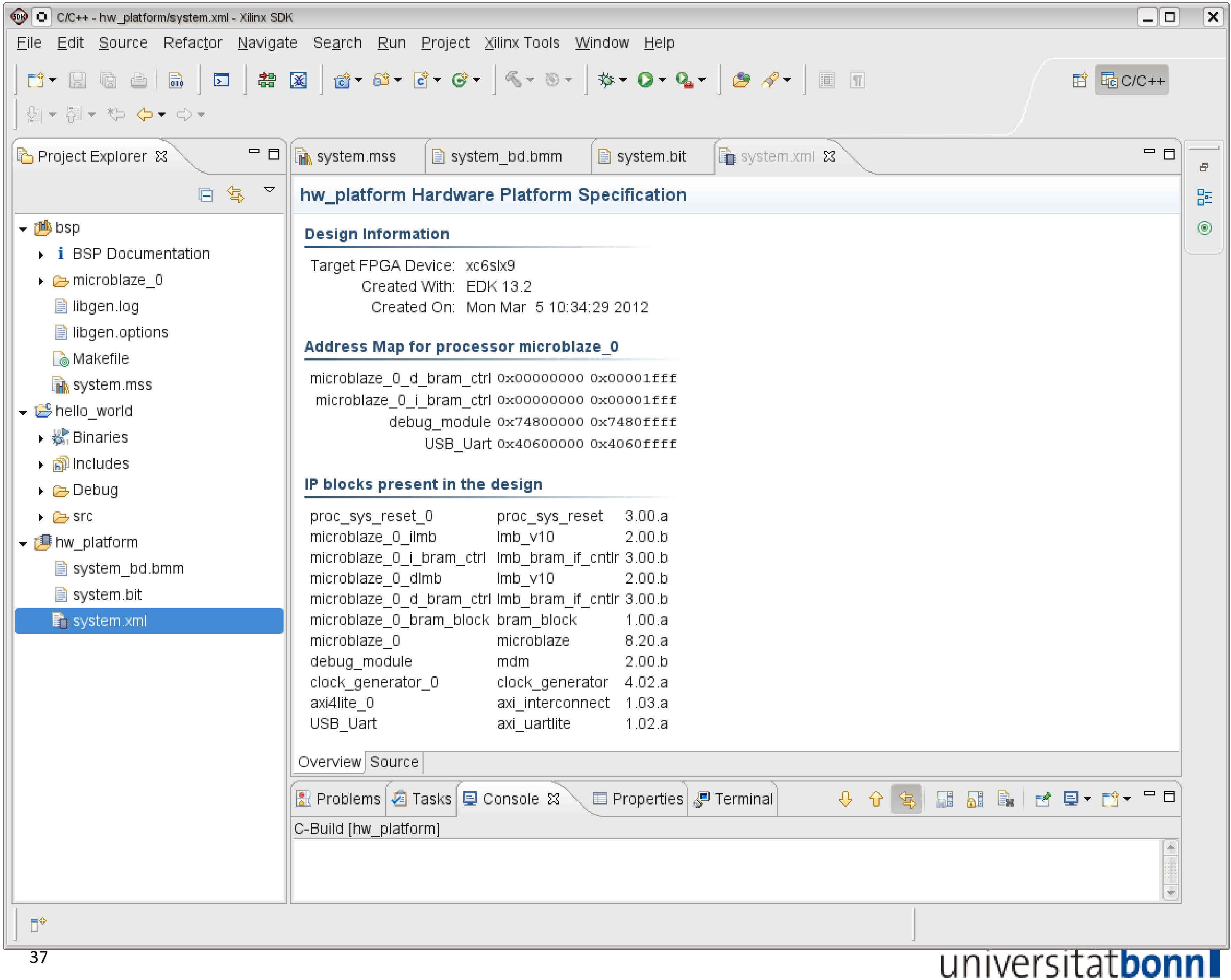
Address Map for processor microblaze_0

```
microblaze_0_d_bram_ctrl 0x00000000 0x00001fff
microblaze_0_i_bram_ctrl 0x00000000 0x00001fff
            debug_module 0x74800000 0x7480ffff
                USB_Uart 0x40600000 0x4060ffff
```

IP blocks present in the design

```
proc_sys_reset_0           proc_sys_reset      3.00.a
microblaze_0_ilmb          lmb_v10             2.00.b
microblaze_0_i_bram_ctrl   lmb_bram_if_cntlr   3.00.b
microblaze_0_dlmb          lmb_v10             2.00.b
microblaze_0_d_bram_ctrl   lmb_bram_if_cntlr   3.00.b
microblaze_0_bram_block    bram_block          1.00.a
microblaze_0               microblaze          8.20.a
debug_module               mdm                 2.00.b
clock_generator_0          clock_generator     4.02.a
axi4lite_0                 axi_interconnect    1.03.a
USB_Uart                   axi_uartlite        1.02.a
```

```
C-Build [hw_platform]
```



hw_platform Hardware Platform Specification

Design Information

Target FPGA Device:	xc6slx9
Created With:	EDK 13.2
Created On:	Mon Mar 5 10:34:29 2012

Address Map for processor microblaze_0

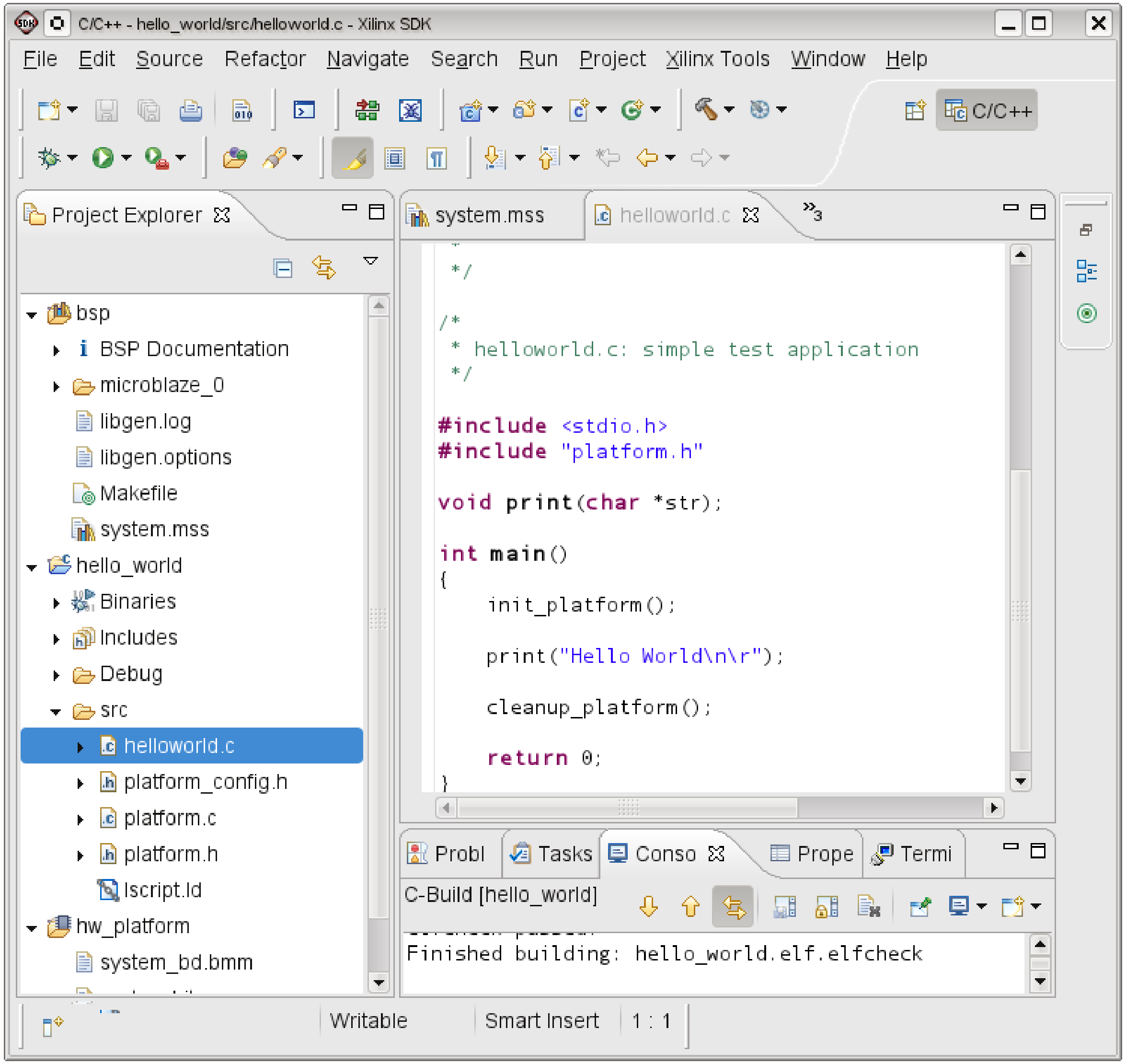
microblaze_0_d_bram_ctrl	0x00000000	0x00001fff
microblaze_0_i_bram_ctrl	0x00000000	0x00001fff
debug_module	0x74800000	0x7480ffff
USB_Uart	0x40600000	0x4060ffff

IP blocks present in the design

proc_sys_reset_0	proc_sys_reset	3.00.a
microblaze_0_ilmb	lmb_v10	2.00.b
microblaze_0_i_bram_ctrl	lmb_bram_if_cntlr	3.00.b
microblaze_0_dlmb	lmb_v10	2.00.b
microblaze_0_d_bram_ctrl	lmb_bram_if_cntlr	3.00.b
microblaze_0_bram_block	bram_block	1.00.a
microblaze_0	microblaze	8.20.a
debug_module	mdm	2.00.b
clock_generator_0	clock_generator	4.02.a
axi4lite_0	axi_interconnect	1.03.a
USB_Uart	axi_uartlite	1.02.a

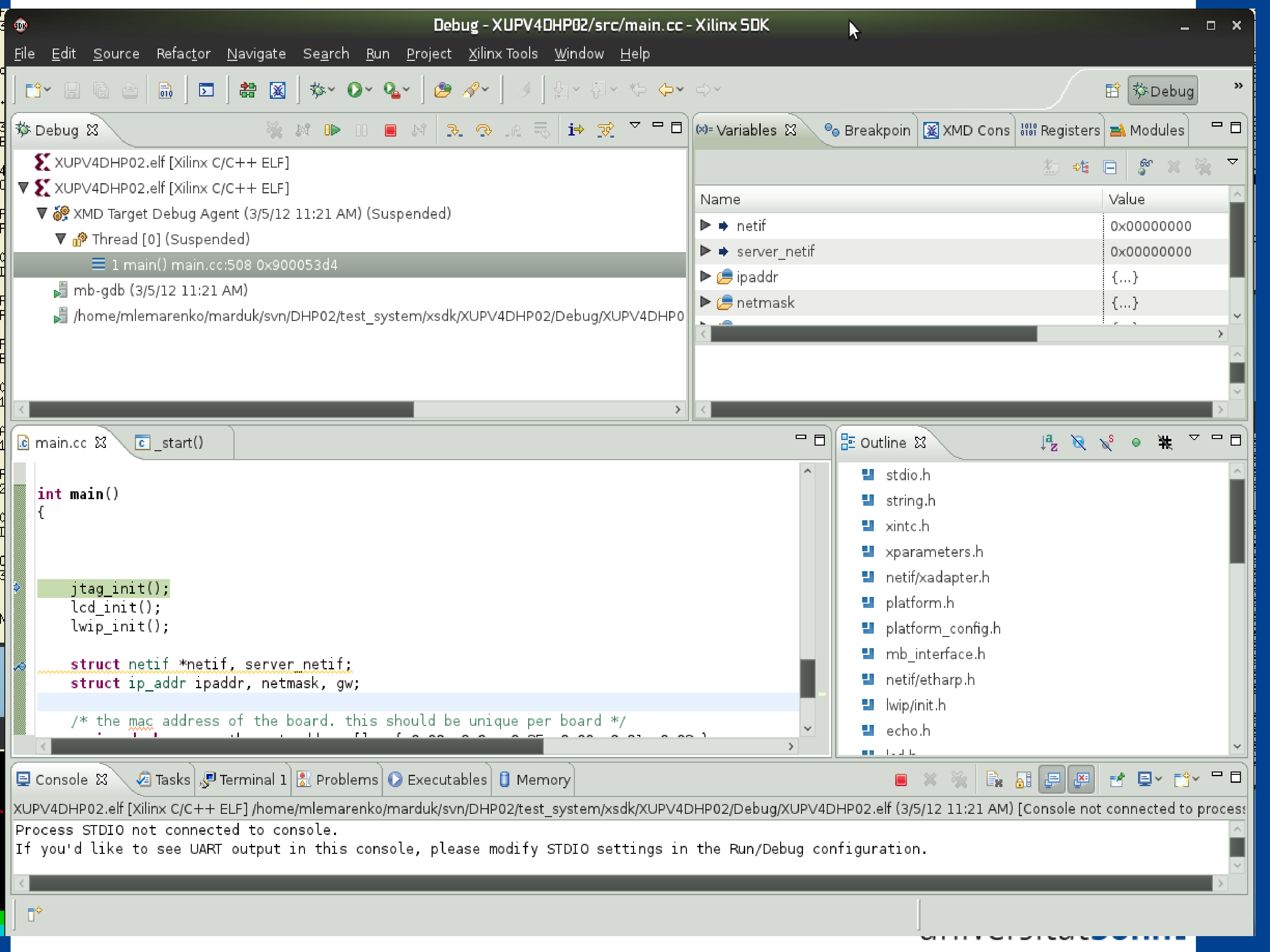
```
C-Build [hw_platform]
```

xSDK – Hallo World Example



```
 */

/*
 * helloworld.c: simple test application
 */

#include <stdio.h>
#include "platform.h"

void print(char *str);

int main()
{
    init_platform();

    print("Hello World\n\r");

    cleanup_platform();

    return 0;
}
```

```
Finished building: hello_world.elf.elfcheck
```



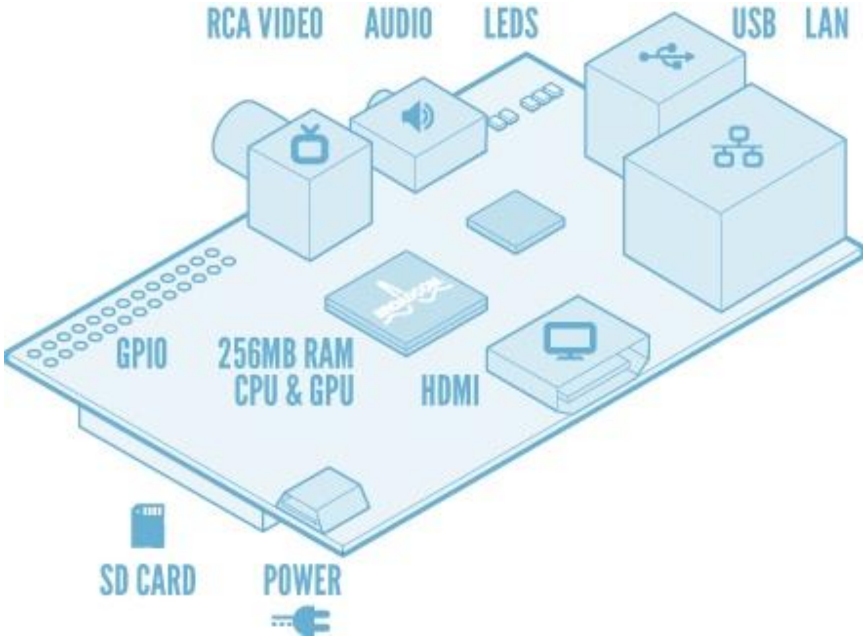
```
XUPV4DHP02.elf [Xilinx C/C++ ELF]
XUPV4DHP02.elf [Xilinx C/C++ ELF]
    XMD Target Debug Agent (3/5/12 11:21 AM) (Suspended)
        Thread [0] (Suspended)
            1 main() main.cc:508 0x900053d4
    mb-gdb (3/5/12 11:21 AM)
    /home/mlemarenko/marduk/svn/DHP02/test_system/xsdk/XUPV4DHP02/Debug/XUPV4DHP0
```

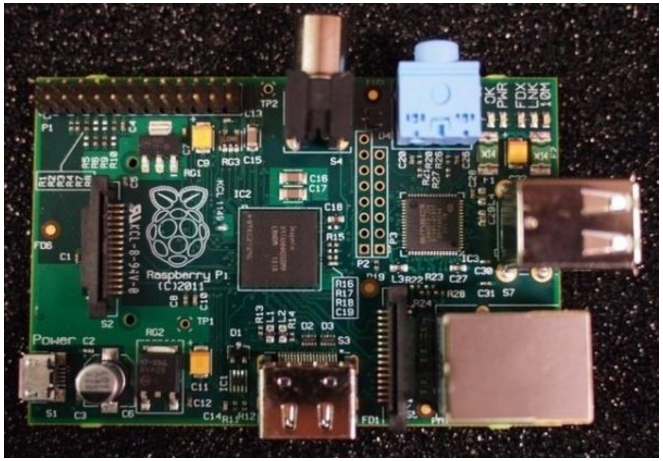
Name	Value
netif	0x00000000
server_netif	0x00000000
ipaddr	{...}
netmask	{...}

```
int main()
{



    jtag_init();
    lcd_init();
    lwip_init();

    struct netif *netif, server_netif;
    struct ip_addr ipaddr, netmask, gw;

    /* the mac address of the board. this should be unique per board */
```

```
XUPV4DHP02.elf [Xilinx C/C++ ELF] /home/mlemarenko/marduk/svn/DHP02/test_system/xsdk/XUPV4DHP02/Debug/XUPV4DHP02.elf (3/5/12 11:21 AM) [Console not connected to process
Process STDIO not connected to console.
If you'd like to see UART output in this console, please modify STDIO settings in the Run/Debug configuration.
```

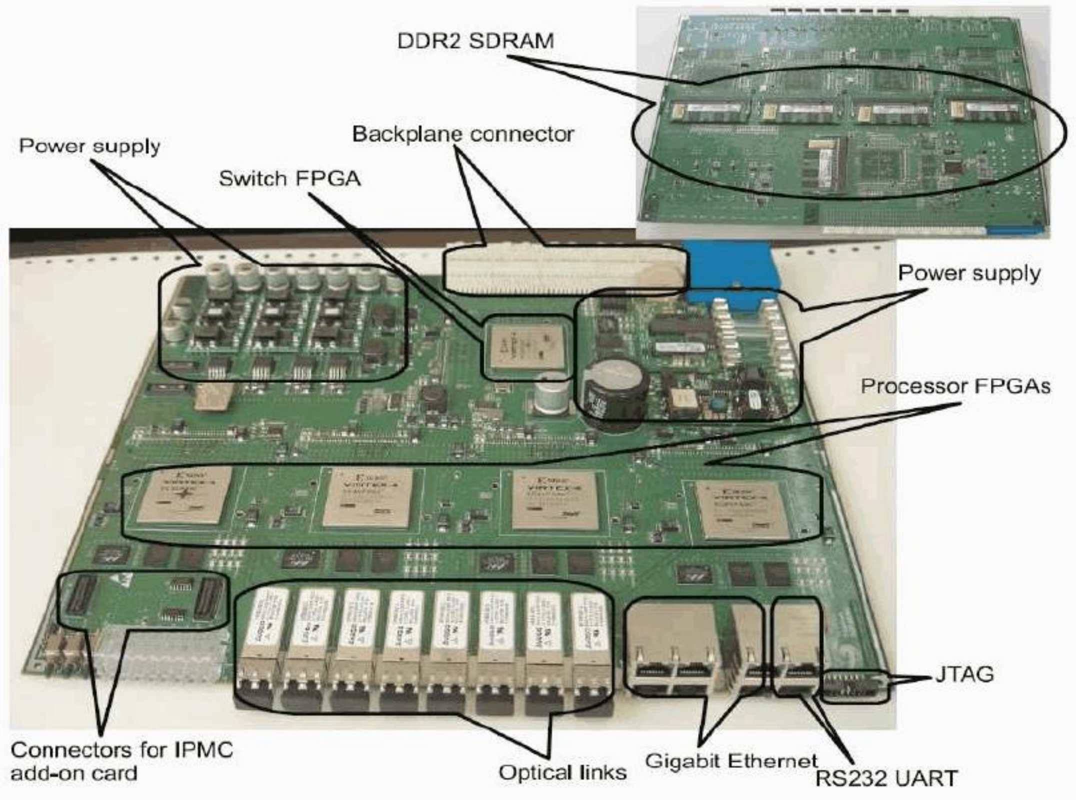
Raspberry Pi for $35





- Broadcom BCM2835 700MHz ARM1176JZFS processor with FPU and Videocore 4 GPU
- GPU provides Open GL ES 2.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile decode
- 256MB RAM
- Boots from SD card, running the Fedora version of Linux
- 10/100 BaseT Ethernet socket
- HDMI socket
- USB 2.0 socket
- RCA video socket
- SD card socket
- Powered from microUSB socket
- 3.5mm audio out jack
- Header footprint for camera connection
- Dimensions: 85.6 x 53.98 x 17mm

Belle2 SXD/PXD Tracking/Triggering



Questions?