

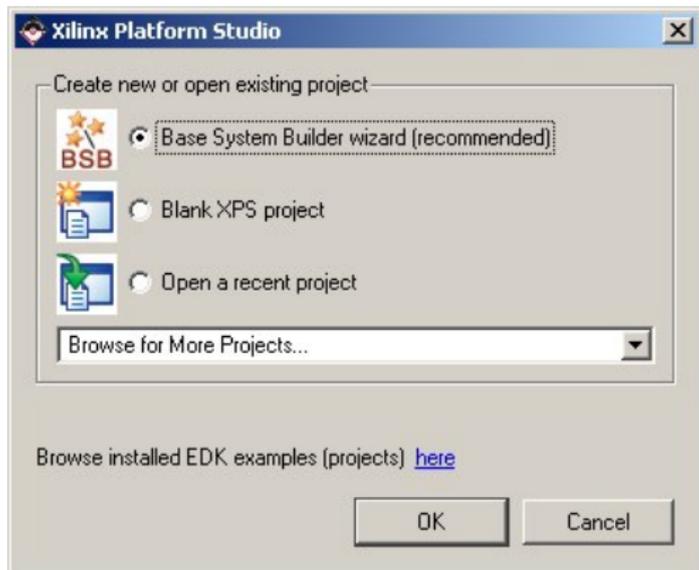


Überblick

- Das Hardwaredesign der einzelnen IPCores erfolgt mit Hilfe der Xilinx ISE
- Die Erzeugung des übergeordneten eingebetteten Designs erfolgt mit Hilfe des Xilinx XST
- Mit dessen Hilfe kann ein Design erzeugt werden, in dem der PowerPC direkt programmiert werden kann
- Die dafür notwendigen Schritte werden im Folgenden dargestellt



Neues Projekt erstellen





Pfade angeben





Neues Design erzeugen





Board auswählen

Base System Builder - Select Board

Select a target development board:

Select board

I would like to create a system for the following development board

Board vendor:

Board name:

Board revision:

Note: Visit the vendor website for additional board support materials.

[Vendor's Website](#) [Contact Info](#)

[Download Third Party Board Definition Files](#)

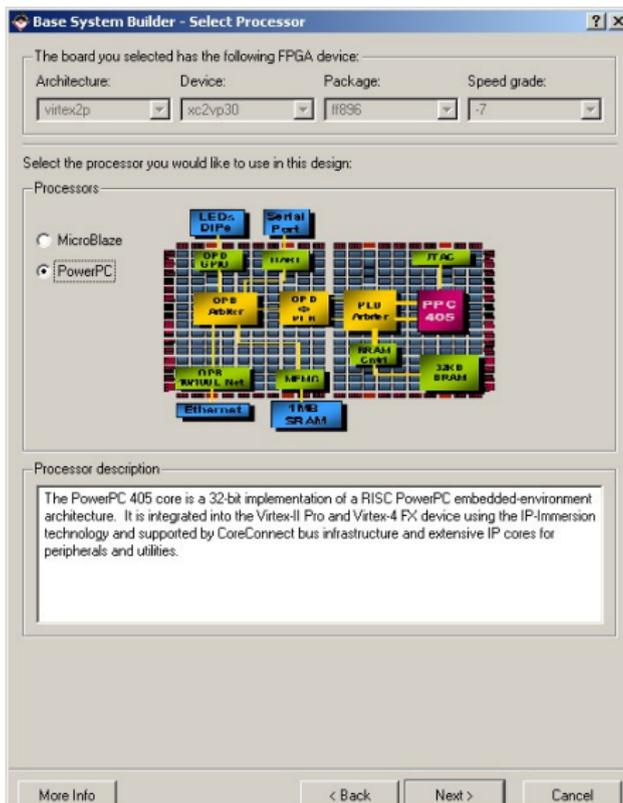
I would like to create a system for a custom board

Board description

The XUP Virtex-II Pro Development System provides an advanced hardware platform that consists of a high performance Virtex-II Pro Platform FPGA surrounded by a comprehensive collection of peripherals that can be used to create a complex system and to demonstrate the capability of the Virtex-II Pro Platform FPGA.

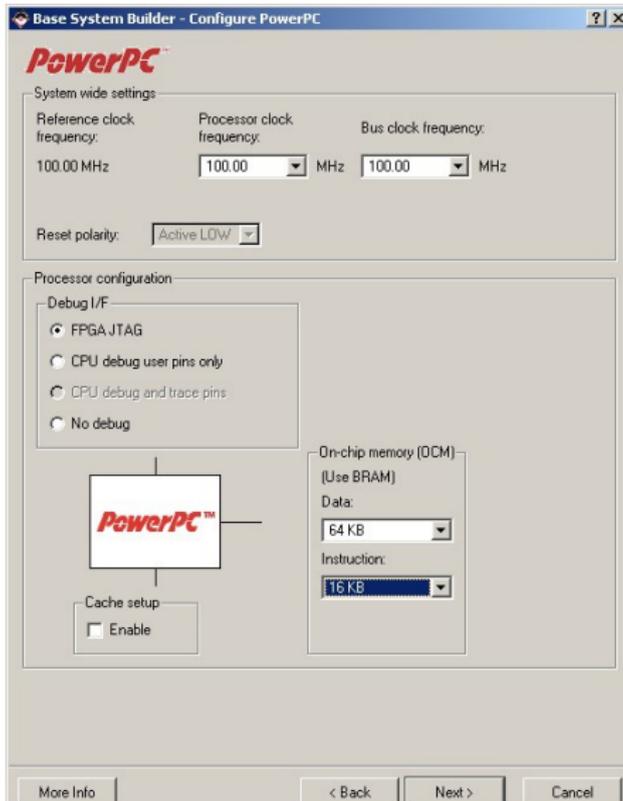


Prozessor auswählen



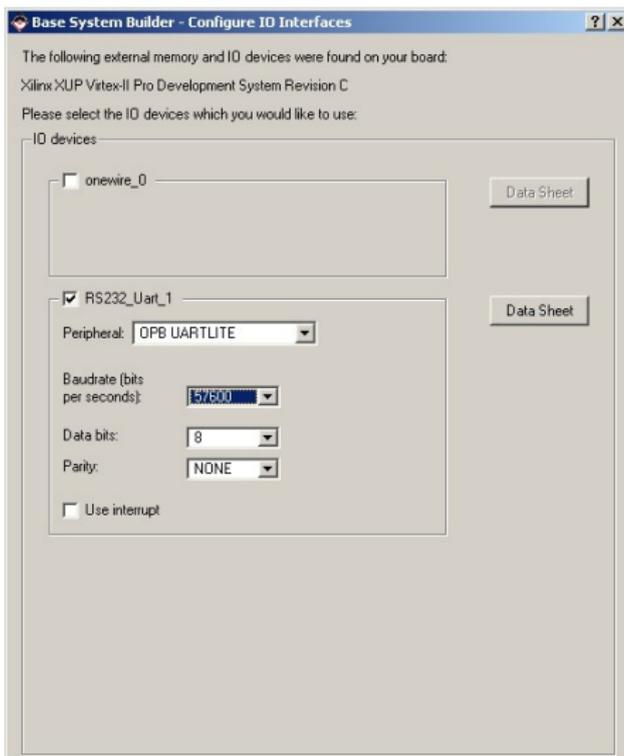


Einstellungen des PowerPC angeben



Konfiguration der Ein-/Ausgabe

Es wird zur Ausgabe die serielle Schnittstelle verwendet:





Konfiguration der Ein-/Ausgabe



Konfiguration der Ein-/Ausgabe

Weitere Module zur Ein-/Ausgabe werden nicht benötigt:





Konfiguration der Ein-/Ausgabe

Gleiches gilt für den Audio-Codec:





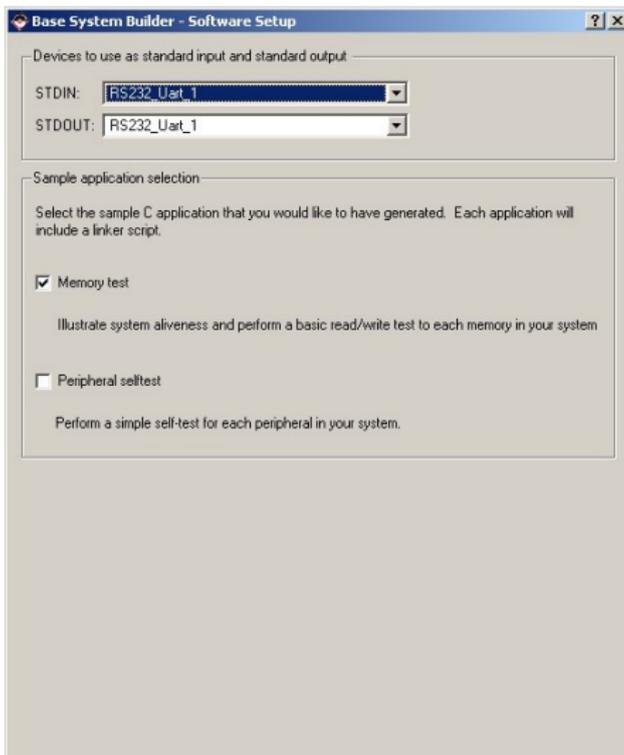
Zusätzliche interne Geräte

Die Einstellungen werden beibehalten:



Software Setup

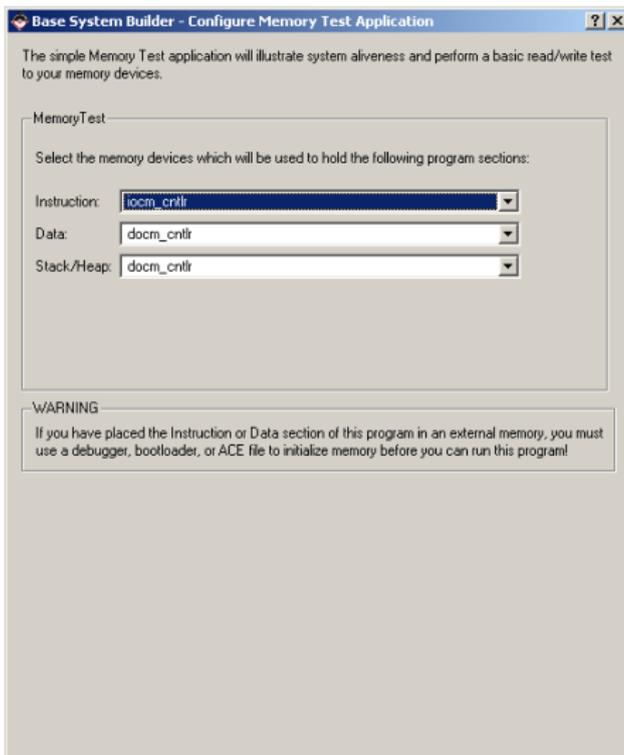
Ein Speicher-Test soll generiert werden:





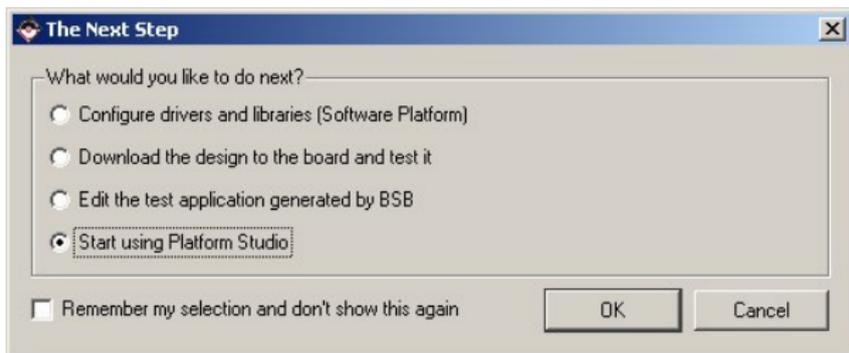
Software Setup

Die Einstellungen werden übernommen:





Start des Plattform-Studios:





Hauptfenster

Xilinx Platform Studio - D:\DIPLOM\progen\tutorial\tutorial.xmp - [System Assembly View1]

File Edit View Project Hardware Software Device Configuration Debug Simulation Window Help

Project Applications IP Catalog

Platform

Project Files

- MHS File: tutorial.mhs
- MSS File: tutorial.mss
- UCF File: ddata/tutorial.ucf
- IMPACT Command File: etc/download...
- Implementation Options File: etc/last_r...
- Bitgen Options File: etc/bitgen.ut

Project Options

- Device: xc2vp30H896-7
- Netlist: TopLevel
- Implementation: XPS
- HDL: VHDL
- Sim Model: BEHAVIORAL

Reference Files

- Log Files
- Report Files

Filters

Bus Interface Ports Addresses Connection Filters

Name	Bus Connection	IP Type	IP Version
ppc405_0		ppc405	2.00.c
ppc405_1		ppc405	2.00.c
iocm		iocm_v10	2.00.a
docm		dsocm_v10	2.00.a
opb		opb_v34	1.02.a
opb		opb_v20	1.10.c
plb2opb		plb2opb_bridge	1.01.a
itagppc_0		itagppc_cntrl	2.00.a
iocm_cntrl		isocm_if_cntrl	3.00.a
docm_cntrl		dsocm_if_cntrl	3.00.a
RS232_Uart_1		opb_uartlite	1.00.b
LEDs_4Bit		opb_gpio	3.01.b
plb_bram_if_cntrl_1		plb_bram_if_cntrl	1.00.b
reset_block		proc_sys_reset	1.00.a
isocm_bram		bram_block	1.00.a
dsocm_bram		bram_block	1.00.a
plb_bram_if_cntrl_1_bram		bram_block	1.00.a
dcm_0		dcm_module	1.00.a

Legend

Master
 Slave
 Master/Slave
 Target
 Initiator
 Connected
 Unconnected

[Platform Studi... System Assembly... BLOCK DIAGRAM]

Reading in the MHS file D:\DIPLOM\progen\tutorial\tutorial.mhs

Block diagram generated.
Block Diagram completed.

Done!

Output Warnings Errors

Fertig



Bearbeiten der Software

Der Speicher-Test dient als Ausgangspunkt für selbst erstellte Software:

The screenshot displays the Xilinx Platform Studio IDE interface. The main window shows a C source file with the following code:

```

35 // Located in: ppc405_0/include/xparameters.h
36 #include "xparameters.h"
37
38 #include "stdio.h"
39
40 #include "xutil.h"
41
42 //=====
43
44 int main (void) {
45
46
47     print("-- Entering main() --\r\n");
48
49     /* Testing BRAM Memory (plb_bram_if_cntlr_1)*/
50     {
51         XStatus status;
52
53         print("Starting MemoryTest for plb_bram_if_cntlr_1:\r\n");
54         print(" Running 32-bit test...");
55         status = XUtil_MemoryTest32((Xuint32*)XPAR_PLB_BRAM_IF_CNTL1_BASEADDR, 512, 0xAAAA5555, XUT_ALLMEM);
56         if (status == XST_SUCCESS) {
57             print("PASSED!\r\n");
58         }
59         else {
60             print("FAILED!\r\n");
61         }
62         print(" Running 16-bit test...");
63         status = XUtil_MemoryTest16((Xuint16*)XPAR_PLB_BRAM_IF_CNTL1_BASEADDR, 1024, 0xAAS5, XUT_ALLMEM);
64         if (status == XST_SUCCESS) {
65             print("PASSED!\r\n");
66         }
67         else {
68             print("FAILED!\r\n");
69         }
70     }
71 }

```

The IDE interface includes a menu bar (File, Edit, View, Project, Hardware, Software, Device Configuration, Debug, Simulation, Window, Help), a toolbar, and a Project Explorer on the left showing a project named "TestApp_Memory". The bottom status bar shows the current file is "TestApp_Memory".



Synthetisierung des Designs

- Das Design wird synthetisiert über *Device Configuration, Create Bitstream*
- Dieser Vorgang dauert ca. 15 Min.
- Zur Anzeige der Ausgabe wird noch eine serielle Verbindung benötigt.



Erstellen einer seriellen Verbindung:

Nach dem Start des Windows Hyperterminal wird eine neue Verbindung angelegt:





Auswahl der Schnittstelle am Rechner:

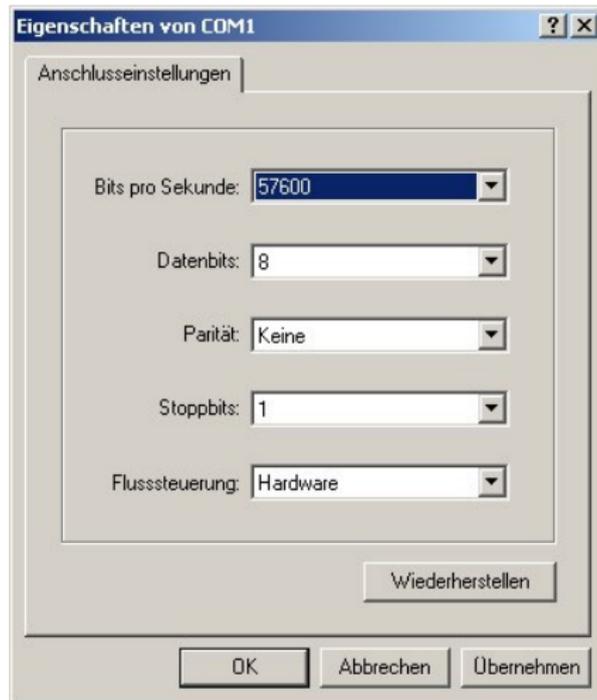
Hier muss der richtige COM-Port ausgewählt werden. Dieser ist abhängig von dem USB-Anschluss, an den das Verbindungskabel angeschlossen wird:





Einstellen der Baud-Rate

Die Baud-Rate muss der entsprechen, die in dem Design definiert wurde:





Hauptfenster Hyper-Terminal

Daten, die von dem erstellten Design in die Standardausgabe geschrieben werden, erscheinen nun im Hauptfenster. Hierbei kann die *print* Funktion der Xilinx-Bibliothek verwendet werden. Sollte das bekanntere *printf* bevorzugt werden, so bietet Xilinx hier mit *xil_printf* eine entsprechende Funktion an, die deutlich weniger Ressourcen benötigt.

