

Entwurf und Test eines Minimalsystems

11. November 2014

Zusammenfassung

In dieser Schritt-für-Schritt-Anleitung wird mit EDK ein minimales Rechnersystem konfiguriert, die erforderlichen Dateien für die Programmierung der Rechnerhardware in den Logikschaltkreis und die Programmeentwicklung erzeugt. Dann wird SDK gestartet, die Projektumgebung eingerichtet, ein kleines Programm zum Ausprobieren der Hardware erstellt und mit diesem der entworfene Rechner ausprobiert.

1 Das zu entwerfende System

Abb. 1 zeigt die Blockstruktur des zu entwerfenden Rechnersystems. Es besteht aus dem Prozessor, einem über lokale Speicherbusse (LMB's) und entsprechende Controller angeschlossenen Dualport-Speicher für die Befehle und Daten, einer über den AXI-Bus angeschlossene Ausgabeschnittstelle für die Ansteuerung der Leuchtdioden auf der Baugruppe, einem über den AXI-Bus angeschlossenen Debug-Controller und einer Rücksetzschialtung. Zum besseren Verständnis sind im Veranstaltungsskript die Abschnitte ## durchzuarbeiten.

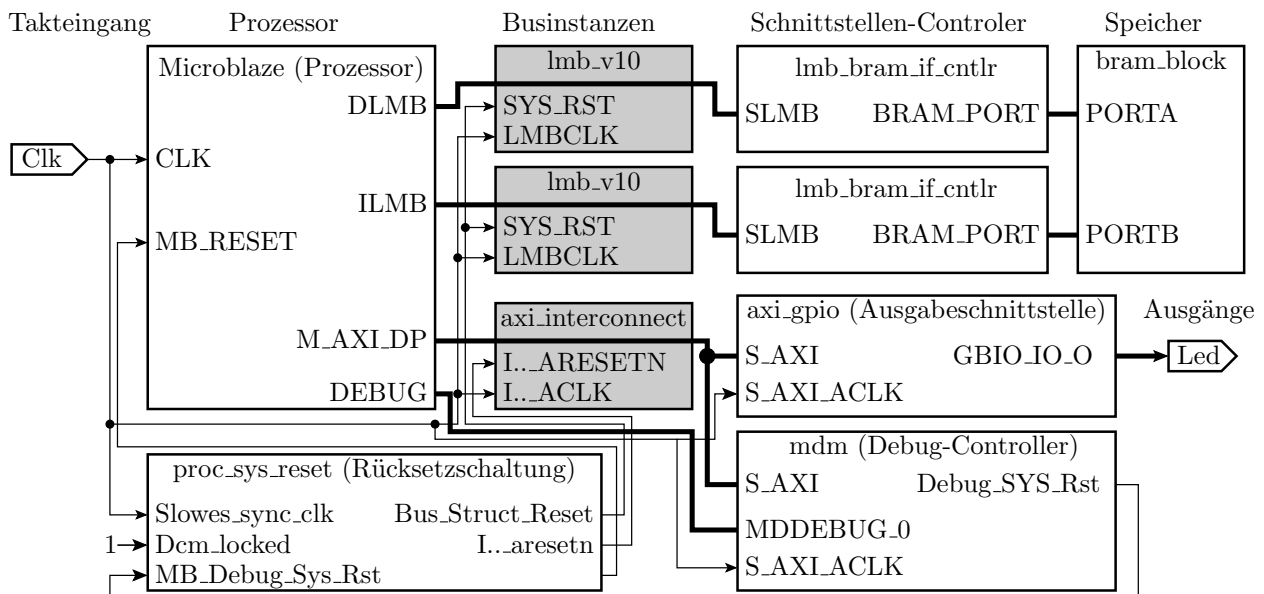


Abbildung 1: Das zu entwerfende Rechnersystem

2 Hardware-Entwurf

Der System Builder von EDK ist in der Lage, das System fast vollständig zu konfigurieren. Der Rest ist manuell nachzubessern. Abb. 2 zeigt die Programmoberfläche von EDK und die auszuwählende Schaltfläche »New BSP-Project« für den Start des System Builders.

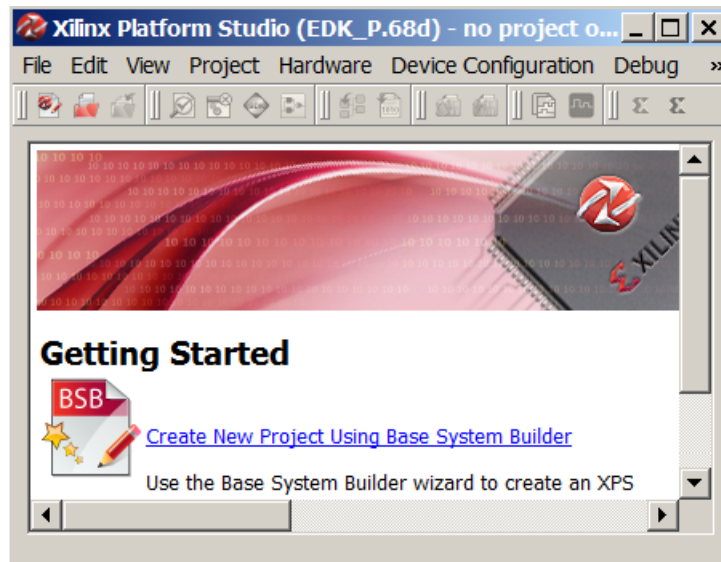


Abbildung 2: Startoberfläche von EDK

In dem sich öffnenden Fenster, Abb. 3, ist die Projektdatei und »AXI-System« auszuwählen. Für jedes neue Projekt sollte ein eigenes Verzeichnis angelegt werden, was auch innerhalb des Menüs möglich ist. Die nicht gezeigten Auswahlfenster in der Abbildung darunter bleiben leer. Weiter mit »Ok«.

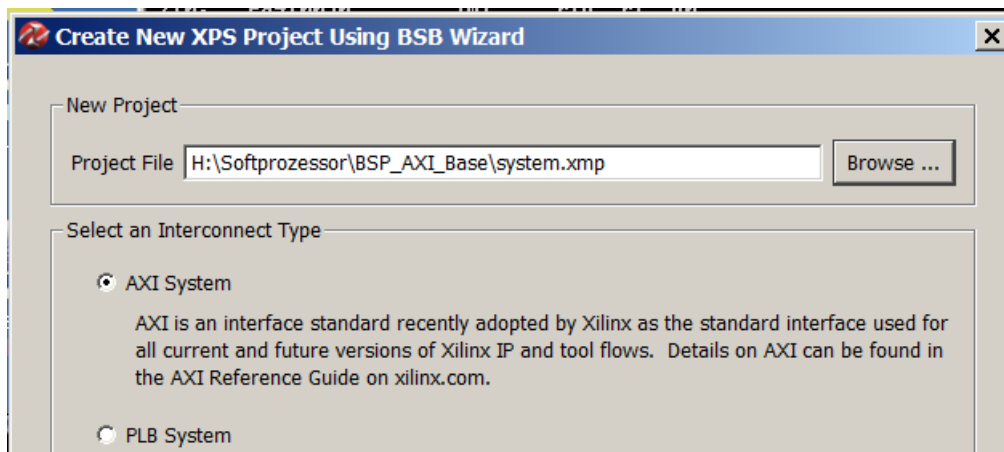


Abbildung 3: Festlegung des Arbeitsverzeichnisses und der Art des Bussystems

Im nächsten Fenster, Abb. 4, sind auszuwählen:

- »Create a System for a Custom Board« (Wir haben keines der auswählbaren Boards),
- der Typ des zu programmierenden Logikschaltkreises (spartan6, xc6slx16, ..),
- die Taktfrequenz 100 MHz, die Art des Rücksetzsignals (high-aktiv).

Für die nicht gezeigten Eintragsfelder sind die vorgegebenen Werte beizubehalten. Weiter mit »Next«.

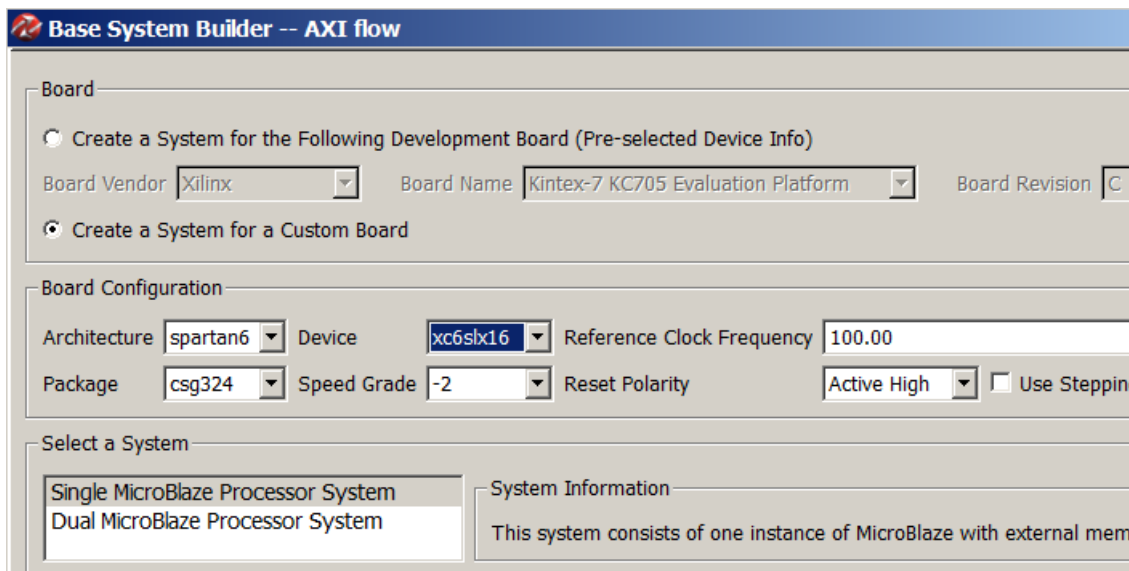


Abbildung 4: Auswahl des Systemtyps und des programmierbaren Logikschaltkreises

Im nächsten Fenster Abb. 5 wird die Blockspeichergröße festgelegt, kann eine Gleitkommaeinheit (FPU) und können periphere Einheiten wie Speichercontroller, Timer, und Schnittstellenschaltungen hinzugefügt werden. Unser Minimalsystem soll nur eine Ausgabe-schnittstelle für die acht Led's auf der Baugruppe erhalten. Das Hinzufügen erfolgt über »Add Device«. Die weiterhin erforderlichen Bausteine (Debug-Controller, Prozessor, ...) werden ohne explizite Auswahl eingefügt.

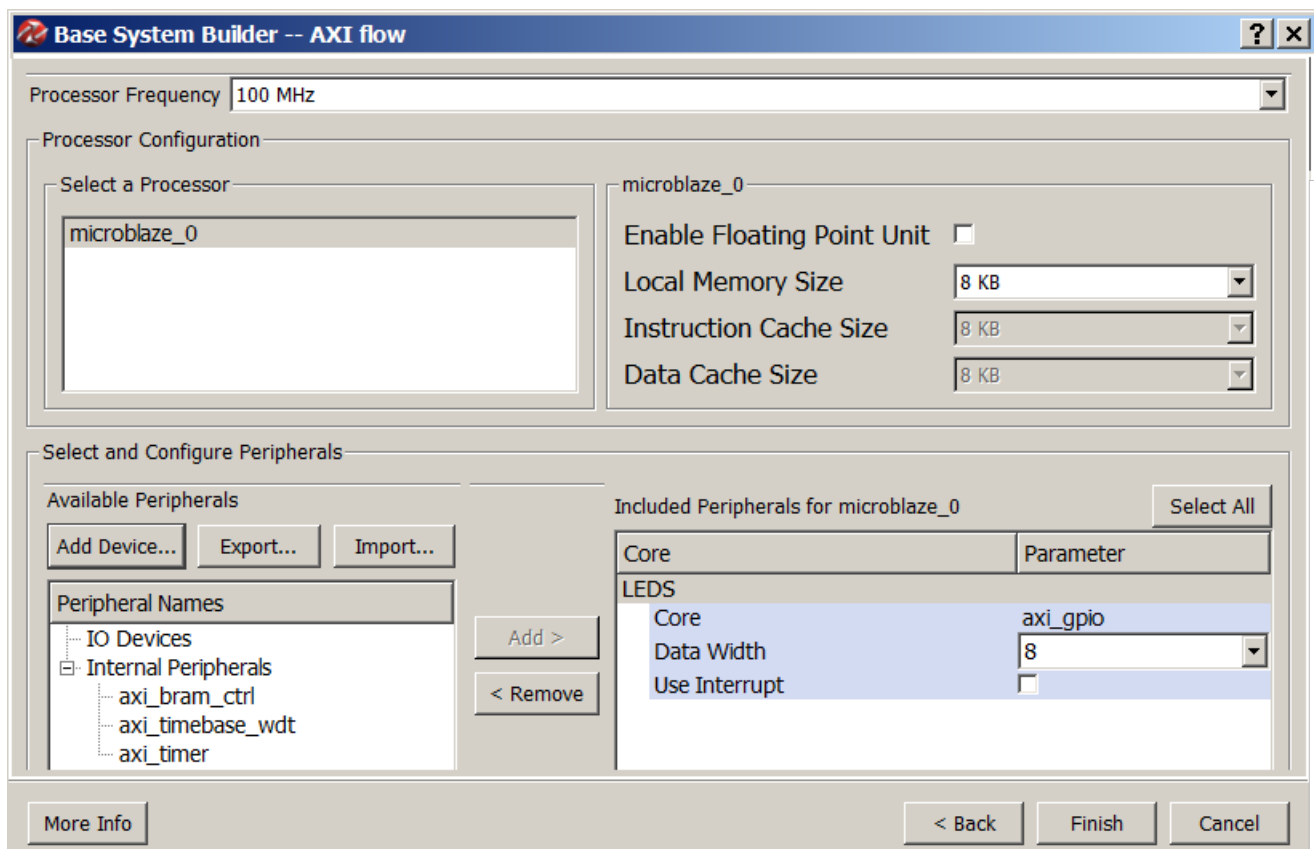


Abbildung 5: Konfiguration des Systems

Nach »Finish« öffnet sich im linken Fenster der »IP Catalog« und rechts die »System

Assembly View / Bus Interfaces« (Abb. 6). In diese Ansicht können weitere Komponenten und Busse aus dem Bausteinkatalog zum Rechnersystem hinzugefügt und verbunden werden. In unserem Projekt hat der »System Builder« bereits alle erforderlichen Bausteile ausgewählt und mit den richtigen Bussen verbunden. Wechsel zum Reiter »Ports».

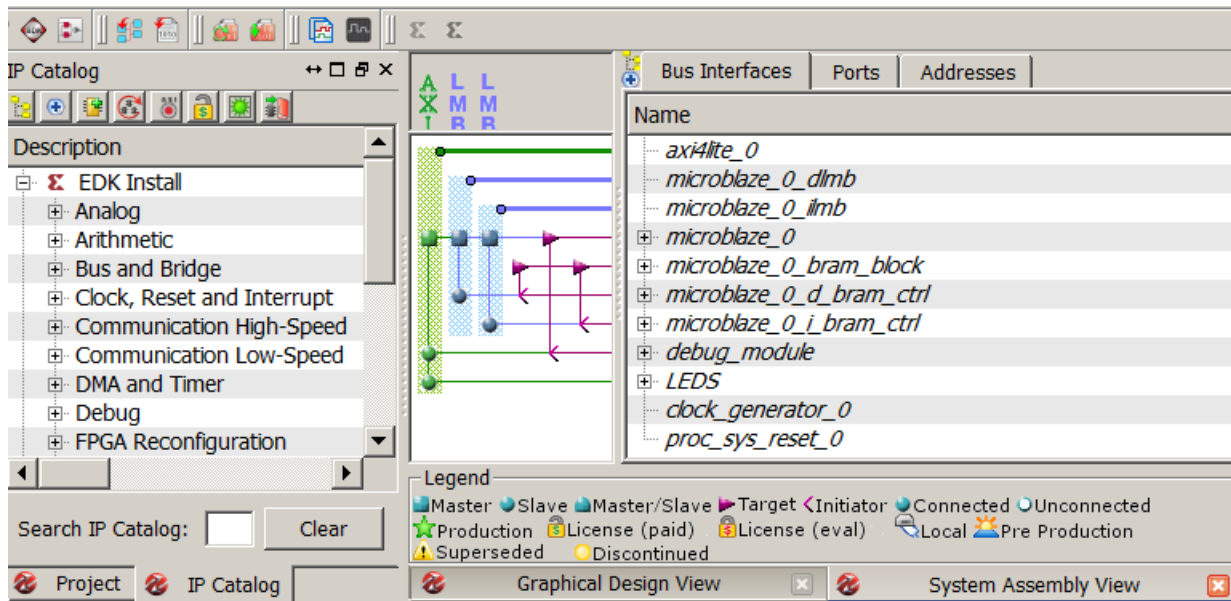



Abbildung 6: Das konfigurierte System

In der Port-Ansicht, Abb. 7, sind die Einträge für die externen Anschlüsse (External Ports) so noch nicht ganz richtig. Unser Rechnersystem benötigt zum Testen keinen Reset-Eingang und der 100MHz-Eingabetakt ist nicht über differentielle Eingänge, sondern nur über einen Eingang an den FPGA angeschlossen. Die Ausgänge zu den Leuchtdioden soll einfach nur »Led« heißen. Der Baustein »clock_generator« ist in unserem Beispiel, in dem die Frequenz des internen Takts gleich der des Eingangstakts ist, eine einfache Verbindung. Dieser Baustein ist ohne Funktion und soll zur Vereinfachung weggelassen werden.

Name	Connected Port	Direction	Range	Class	Frequency(Hz)
External Ports					
CLK_N	clock_generator_0::CLKIN	I		CLK	100000000
CLK_P	clock_generator_0::CLKIN	I		CLK	100000000
LEDS_TRL_O	LEDS::[gpio_0]::GPIO_IO_O	O	[0:7]	NONE	
RESET	proc_sys_reset_0::Ext_Reset_In	I		RST	
	clock_generator_0::RST				
axi4lite_0					
microblaze_0_dmb					
microblaze_0_ilm					
microblaze_0					
microblaze_0_bram_block					
microblaze_0_d_bram_ctrl					
microblaze_0_i_bram_ctrl					
debug_module					
LEDS					
clock_generator_0					
proc_sys_reset_0					

Abbildung 7: Port-Ansicht nach der Konfiguration mit dem »System Builder«

Der einfachste Weg zur Nachbesserung ist, die externen Signale »CLK_N«, »CLK_P« und »Reset« sowie den die Taktversorgung »clock_generator_0« zu löschen, die Ausgänge

»LEDS_TRI_O« in »Led« umzubenennen und die dann fehlenden Verbindungen neu zuzuordnen. Löschen erfolgt durch Mausklick auf den Eintrag und »Delete external Port« bzw. »Delete Instance and all its connections« und erfordert etwas Übung und Geduld. Der neue Taktanschluss wird über einen Klick auf die Schaltfläche  oben rechts erzeugt. In das aufgehende Menü in Abb. 8 ist der Anschlussname, hier »Clk«, die Signalfflussrichtung, hier »I« für Eingang/Input, und der Signaltyp, hier »CLK« für Takt einzutragen. Nach »Ok« ist in der Zeile für den Takteintrag noch die Frequenz zu ergänzen (siehe Abb. 9).

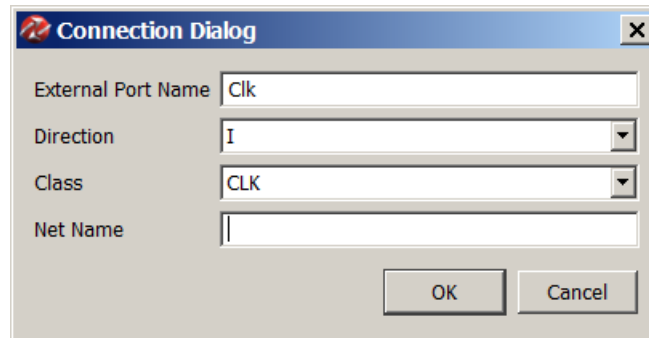


Abbildung 8: Dialogfenster für das Anlegen externer Anschlüsse

Namensänderungen erfolgen durch Mausklick auf den alten Namen. Auch das erfordert etwas Übung und Geduld. Abb. 9 zeigt die fertig geänderten Einträge für die externen Anschlüsse.

Name	Connected Port	Direction	Range	Class	Frequency(Hz)	Differential Polarity
External P...						
Clk		I		CLK	100000000	
Led	LEDS::[gpio_0]::GPIO_IO_O	O	[0:7]	NONE		

Abbildung 9: Geänderte Einträge für die externen Anschlüsse

Das Entfernen der Taktversorgung und der externen Anschlüsse hinterlässt offene Eingänge an Bausteinen und entsprechende Fehlermeldungen beim »Design Rule Check« (Run DRCs). An alle offenen Takteingänge (Eingänge der Klasse »CLK«) ist der externe Takt anzuschließen. In Abb. 10 ist für den Takteingang des AXI-Busses das Auswahlménü für die Zuordnung des Taktanschlusses geöffnet. Am Datenbus »microblaze_0_dmb« darunter ist das Namensfeld für den Taktanschluss noch leer, d.h. der Takteingang ist noch offen.

Name	Connected Port	Direction	Range	Class	IP Type	Diffe	Sensitivity
External Ports							
Clk		I		CLK			
Led	LEDS::[gpio_0]::GPIO_IO_O	O	[0:7]	NONE			
axi4lite_0					axi...		
interconnect_...	External Ports			Clk			
INTERCONNE...	proc_sys_reset_0:						
microblaze_0_dmb							
LMB_CLK		I		CLK			
SYS_RST	proc_sys_reset_0::Bus_Struct_Re...	I		RST			
microblaze_0_ilm					lm...		

Abbildung 10: Neuordnung des Takts

Insgesamt muss der Takt neunmal angeschlossen werden. Die angeschlossenen Einheiten und ihre Eingabetakte erscheinen auch als »Connected Ports« in der Beschreibungszeile des externen Takteingangs (Abb. 11).

Name	Connected Port	Direction	Range	Class
External Ports				
Clk	proc_sys_reset_0::Slowest_sync_clk			
	microblaze_0_ilmc::LMB_CLK			
	microblaze_0_i_bram_ctrl::[SLMB]::LMB_Clk			
	microblaze_0_dlmc::LMB_CLK			
	microblaze_0_d_bram_ctrl::[SLMB]::LMB_Clk	I		CLK
	microblaze_0::[DLMB:ILMB:M_AXI_DP:M_AXI_IP]::CLK			
	debug_module::[S_AXI]::S_AXI_ACLK			
	axi4lite_0::[S_AXI_CTRL]::INTERCONNECT_ACLK			
LEDS::[S_AXI]::S_AXI_ACLK				

Abbildung 11: «Connected Ports» am Takteingang

An der Reset-Schaltung müssen außer dem Takt zwei weiteren Eingängen Signalquellen zugeordnet werden. In einem System ohne Reset-Taste ist das Signal »Ext_Reset_in« mit dem logischen Wert null (net_gnd) »Reset inaktiv« und der Eingang »DCM_lock« mit dem logischen Wert eins (net_vcc) »Taktversorgung synchronisiert« zu verbinden (Abb. 12). Wenn hier ein Fehler gemacht wird, bleibt der Prozessor dauerhaft rückgesetzt, so dass sich am Ende des Software-Entwurfs kein Programm starten lässt.

Name	Connected Port	Direction	Range	Class
proc_sys_reset_0				
Slowest_sync_clk	External Ports::Clk	I		CLK
Ext_Reset_In	net_gnd	I		RST
Aux_Reset_In		I		RST
MB_Debug_Sys_Rst	debug_module::Debug_SYS_Rst	I		RST
Dcm_locked	net_vcc	I		
MB_Reset	microblaze_0::MB_RESET	O		RST
BUS_STRUCT_RESET	microblaze_0_ilmc::SYS_RST	O		RST
	microblaze_0_dlmc::SYS_RST	O		
Peripheral_Reset		O		RST
Interconnect_aresetn	axi4lite_0::INTERCONNECT_ARESETN	O		RST
Peripheral_aresetn		O		RST

Abbildung 12: Anschlüsse an der Reset-Schaltung

Über dem rechten großen Fenster für die Systembeschreibung gibt es außer »Bus Interfaces« und »Ports« einen dritten Reiter »Addresses«. Über diesen Reiter werden den Speichern und den Registern der peripheren Bausteine Adressbereiche zugeordnet (Abb. 13). Die automatisch generierten Einstellungen sind in Ordnung und verlangen keine Nachbesserung.

Instance	Base Name	Base Address	High Address	Size	Bus Interface(s)	Bus Name	Lock
microblaze_0's Address Map							
microblaze_0_d_bram_ctrl	C_BASEADDR	0x00000000	0x00001FFF	8K	SLMB	microblaze_0...	<input type="checkbox"/>
microblaze_0_i_bram_ctrl	C_BASEADDR	0x00000000	0x00001FFF	8K	SLMB	microblaze_0...	<input type="checkbox"/>
LEDS	C_BASEADDR	0x40000000	0x4000FFFF	64K	S_AXI	axi4lite_0	<input type="checkbox"/>
debug_module	C_BASEADDR	0x41400000	0x4140FFFF	64K	S_AXI	axi4lite_0	<input type="checkbox"/>

Abbildung 13: Adresszuordnung

Im nächsten Schritt ist die ucf-Datei zu vervollständigen. Dazu ist unter dem Fenster mit dem IP-Katalog der Reiter »Project« auszuwählen und im Baum »Project Files« die ucf-Datei zu öffnen (Abb. 14). In der automatisch generierten Datei system.ucf steht nur drin, dass der Takt eine Frequenz von 100 MHz hat. Die Zuordnungen der Gehäuseanschlüsse darunter fehlen

und sind selbst einzutragen. Die Pinzuordnungen der in der Abbildung fehlenden Einträge für die Leuchtdioden 3 bis 7 sind in der »nexys3.ucf« auf der Web-Seite zu finden. Wichtig ist, dass die Anschlusnamen in der ucf-Datei mit denen der externen Anschlüsse des Rechnersystems übereinstimmen. Zuordnungsfehler bewirken, dass Schaltungsanschlüsse mit falschen Gehäuseanschlüssen verbunden werden. Das wird zum Teil erst beim Programmstart oder der Programmabarbeitung sichtbar.

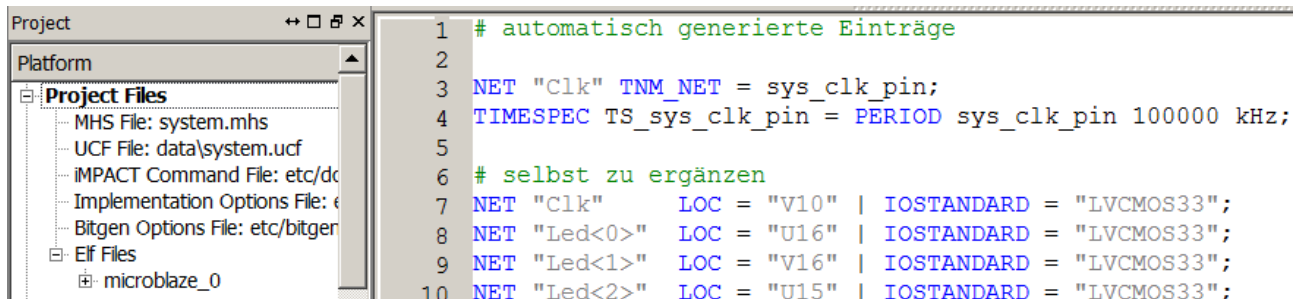


Abbildung 14: Navigator, Projektdateibaum und ucf-Datei

Die fertige Hardware-Beschreibung steht in der Datei »system.mhs« im Project-File-Baum direkt über der ucf-Datei und kann geöffnet, gelesen und editiert werden (Abb. 15). Von dem gesamten bisherigen Entwurf genügt es, die mhs- und die ucf-Datei aufzubewahren (siehe Anleitungen für die Hardware-Erweiterung des Minimalsystems).

Links neben den »Project Files« befinden sich die Icons der Werkzeuge für die Schaltungserzeugung aus der bis hierher erstellten Konfigurationsbeschreibung. Mit »Run DRLs« wird der »Design Rule Check« gestartet. Dieser kontrolliert die Konfiguration des Rechnersystems auf Zulässigkeit. Er erkennt z.B. offene Takt- und Reset-Eingänge. Sind alle mit »Run DRLs« erkennbaren Fehler beseitigt und wurde für alle möglicherweise noch vorhandenen Warnungen kontrolliert, das sie als ignorierbar gelten¹, kann der Generierungsprozess mit »Generate Netlist«, »Generate BitStream« und »Export Design« fortgeführt und abgeschlossen werden.

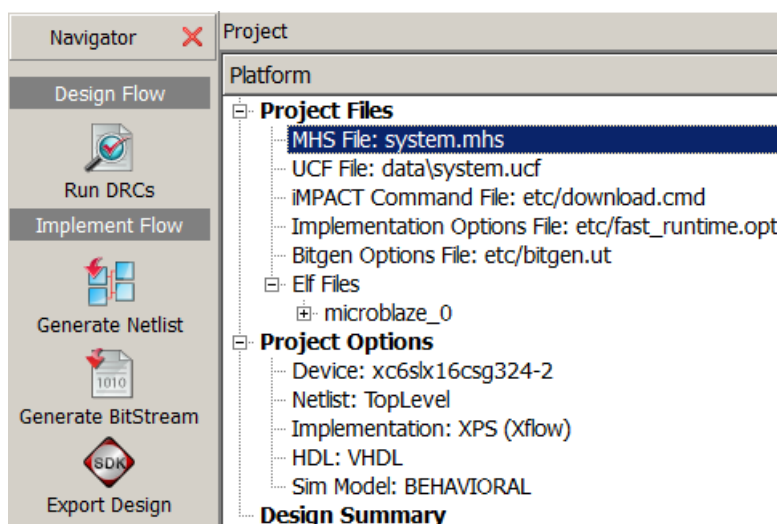


Abbildung 15: Navigator, Projektdateibaum und ucf-Datei

Die Werkzeuge müssen nicht einzeln aufgerufen werden. »Export Design« mit angekreuztem »Include bitstream and ...« startet alle Bearbeitungsschritte nacheinander (Abb. 16). Mit »Ex-

¹Ignorierte Warnungen bei »Run DRLs« sind oft Ursache für Fehlfunktionen bei der späteren Inbetriebnahme und beim Softwaretest. Es empfiehlt sich, dazu auffindbare FAQ-Seiten zu studieren. Gefahrenlos ignorierbar sind in der Regel Warnungen, das Lizenzen abgelaufen oder Ausgänge ungenutzt sind.

port only« werden die drei Ergebnisdateien des Hardware-Entwurfs für den Software-Entwurf nur gespeichert bzw. überschrieben. »Export & Launch SDK« startet zusätzlich die Software-Entwicklungsumgebung SDK.

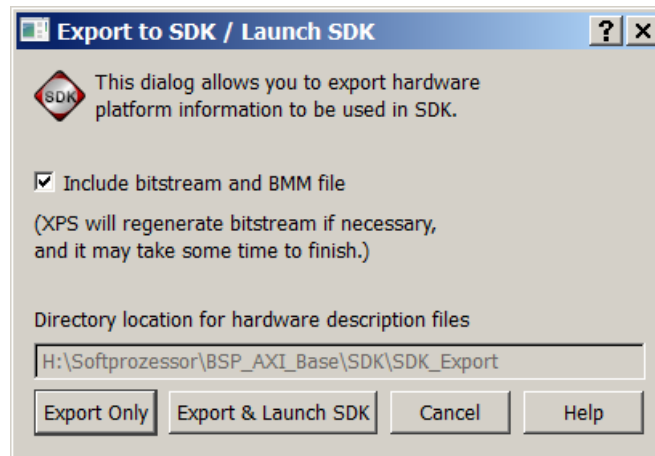


Abbildung 16: Auswahlfenster für den Export

Beim Export und beim Start von SDK ist der Pfad für das Software-Projekt anzugeben (Abb. 17). Das sollte bei einem neuen Projekt ein neu angelegtes leeres Verzeichnis sein.

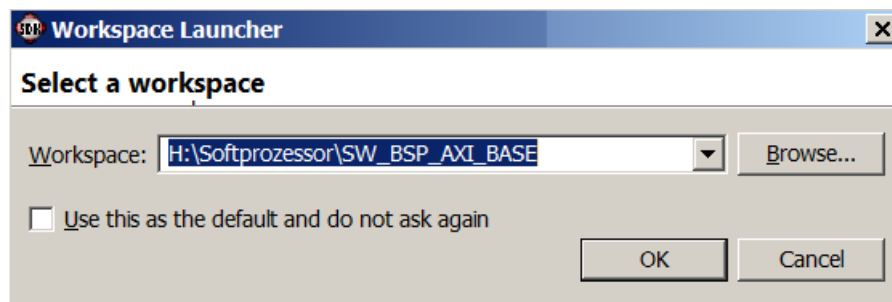


Abbildung 17: Festlegung des Pfads für das Software-Projekt

SDK hängt sich, wenn es so gestartet wird, gern mit einer Java-Fehlermeldung auf. Dann ist entweder »Export & Launch SDK« noch einmal aufzurufen, aber ohne den Haken bei »Include bitstream and ...« oder SDK ist direkt zu starten. Abb. 18 zeigt das Icon von SDK und den Pfad, unter dem es im Startmenü

Start ▷ All Programms

zu finden ist. Beim direkten Start ist der in Abb. 17 festgelegte Workspace auszuwählen.

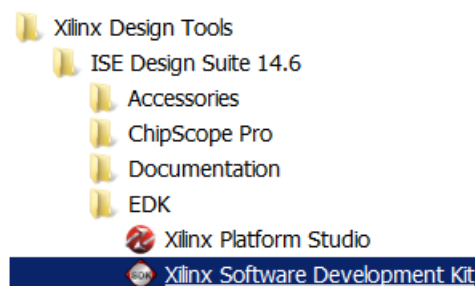


Abbildung 18: Icon und Pfad für den direkten Start von SDK

3 Überspringen des Hardware-Entwurfs

Das Ergebnis des Hardware-Entwurfs steht in drei Dateien:

- bit-Datei: Download-File zur Programmierung der Hardware-Funktionalität
- bmm-Datei: Informationen für den Download des Programms und
- xml-Datei: Informationen, die die Treiber über die Hardware brauchen.

Für den im vorherigen Abschnitt beschriebenen Entwurf stehen diese drei Dateien auf der Webseite und sind in ein Verzeichnis auf dem Rechner, z.B. »Download« herunterzuladen. Das Entwurfssystem SDK ist normal zu starten und in dem sich öffnenden Fenster genau wie in Abb. 17 ein neues leeres Verzeichnis als »workspace« auszuwählen. Die Auswahl der »Hardware Platform Specification« erfolgt über

File ▷ New ▷ Project

und dann weiter mit dem Auswahlfenster Abb. 19 und »Next«.

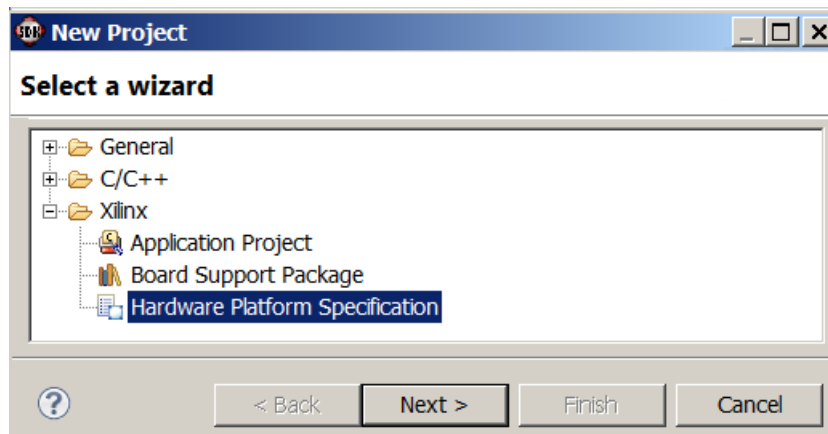


Abbildung 19: Festlegung des Pfads für das Software-Projekt

Im nächsten Fenster, Abb. 20, ist die xml-Datei auszuwählen. Die Auswahl von »Bitstream and BMM Files« darunter, erfolgt, wenn sie im selben Verzeichnis stehen, automatisch. Nach »Finish« wird der Objektbaum für die »Hardware Platform Specification« in derselben Weise wie bei einem Export aus EDK eingerichtet.

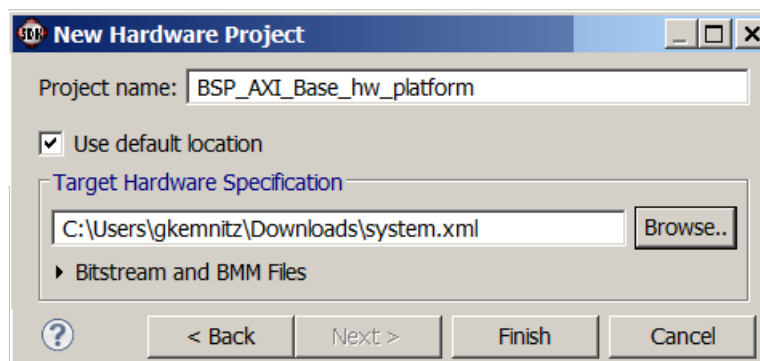


Abbildung 20: Auswahl der Dateien zur Beschreibung der Hardware

4 Software-Entwurf

Nach »Export & Launch SDK« aus EDK bzw. nach »Finish« im Abschnitt zuvor enthält der Objektbaum im »Project Explorer« in Abb. 21 den Teilbaum mit den Dateien zur Beschreibung der Hardware: das bit-File zur Programmierung der Hardware-Funktionalität, das xml-File mit den Hardware-Adressen und Konstanten für die Treiber und das bmm-File, in dem steht, aus welchen Block-RAMs der Befehls- und Datenspeicher zusammengesetzt ist.

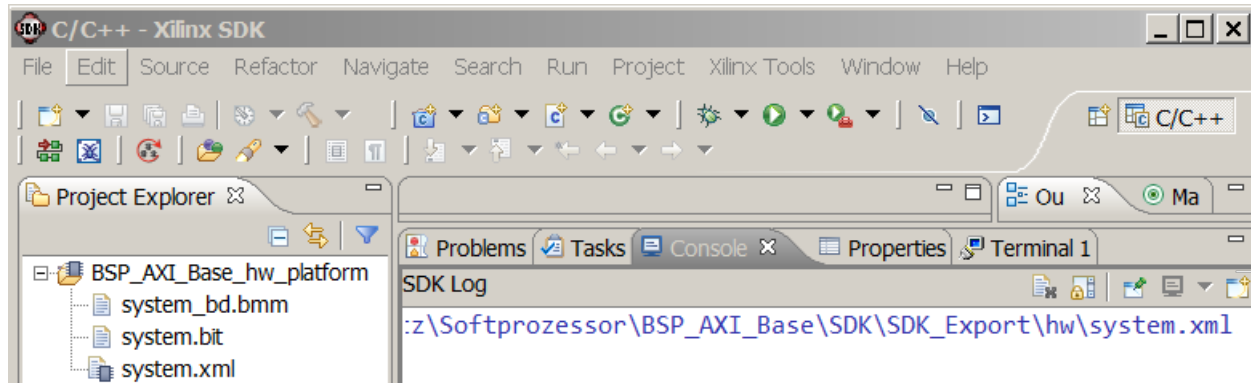


Abbildung 21: Projektbaum mit der »Hardware Platform Specification«

Vor dem Entwurf des ersten Anwendungsprogramms ist ein »Board Support Package« mit den hardware-nahen Funktionen und Treibern zu erzeugen. Über

File ▷ New ▷ Board Support Package

öffnet sich das Fenster in Abb. 22. In ihm ist »standalone« auszuwählen (Entwicklung von Anwendungsprogrammen, die allein ohne Betriebssystem auf dem Prozessor laufen). Im nächsten Fenster ist keine der angebotenen Bibliotheken auszuwählen. Abschluss mit »Ok«.

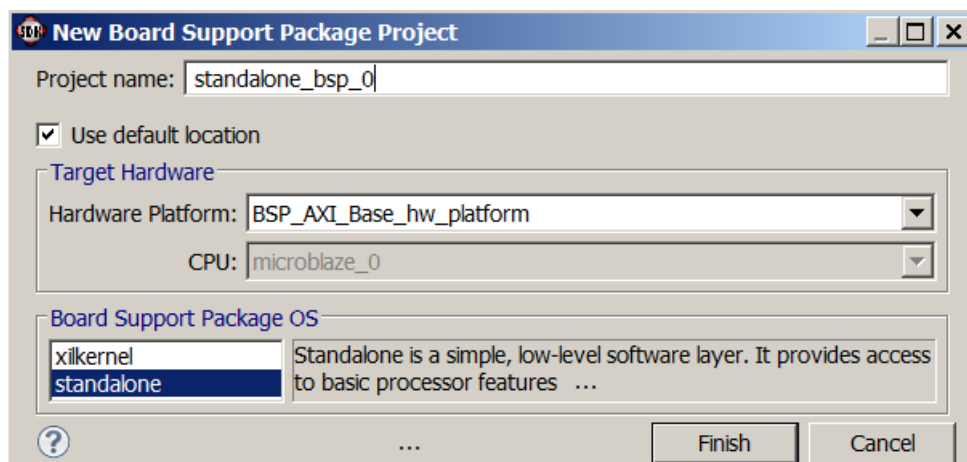


Abbildung 22: Einrichtung des Board Support Packages

Abb. 23 zeigt den Objektbaum mit dem erzeugten »Board Support Package«. Für die Entwicklung der Anwendungsprogramme sind vor allem die Header im Include-Verzeichnis des Prozessors wichtig. Die hardware-spezifischen Adressen und Konstanten sind hier alle als Makros im Header »xparameters.h« zu finden. »XPAR_LEDS_BASEADDR« mit dem Wert 0x400000000000 ist z.B. die Ausgabeadresse an die Leuchtdioden.

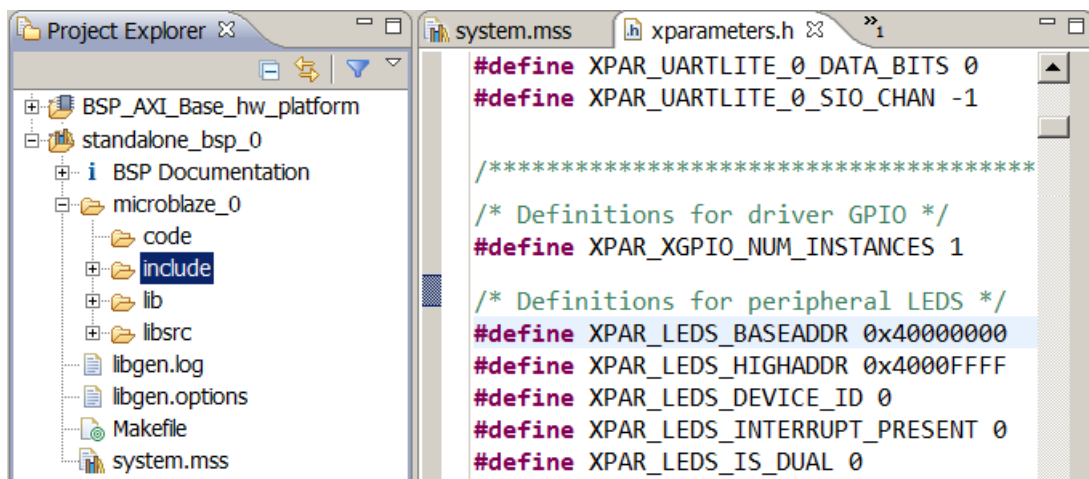


Abbildung 23: Projektbaum mit Board Support Package und der Header »xparameters.h«

Nach all den vorbereitenden Schritten kann nun endlich die eigentliche Software-Entwicklung beginnen. Die Einrichtung eines Software-Projekts erfolgt über:

File ▷ New ▷ Application Project

In dem sich öffnenden Fenster, Abb. 24, ist dem Projekt ein Name zu geben, die Programmiersprache festzulegen sowie eine »Hardware Plattform« und ein »Board Support Package« zuzuordnen. Dann geht es mit »Next« zum nächsten Menü.

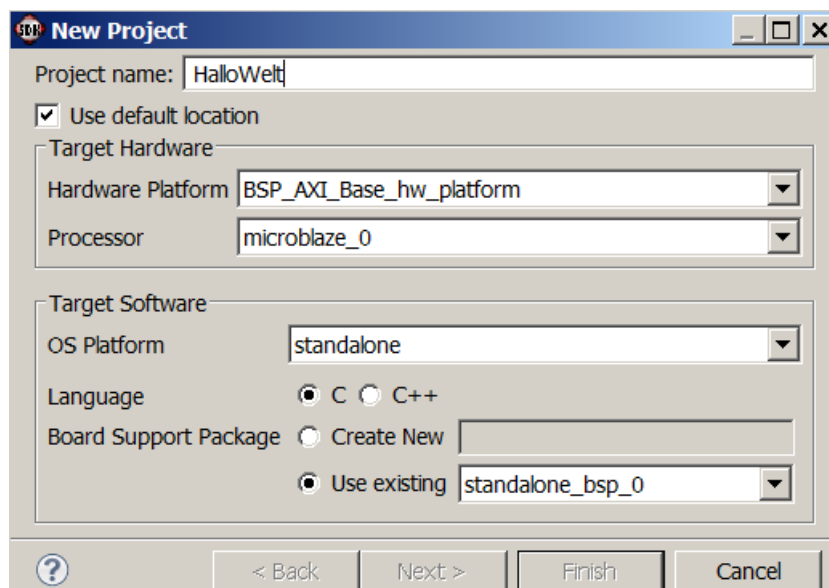


Abbildung 24: Einrichten eines Software-Projekts

Im nächsten Menü, Abb. 25, wird als Vorlage »Hello World« ausgewählt und das Anlegen des »Applikation Projects« mit »Finish« abgeschlossen. Ergebnis ist der dritte Projektbaum in Abb. 26 mit dem Namen des »Application Projects« und dem »helloworld«-Programm.

Das originale Helloworld-Programm schreibt auf die Standardausgabe den Text »Hello World« und beendet sich. Die Zeile mit der Ausgabe der Konstanten 0xB3 an die Leuchtdioden ist manuell eingefügt. »Xio_Out8()« ist im Header »xio.h« und die Ausgabeadresse »XPAR_LEDS_BASEADDR« in »xparameters.h« definiert. Das Übersetzen erfolgt beim Speichern. Bei der Fehlermeldung, dass der Übersetzer »XPAR_LEDS_BASEADDR« oder

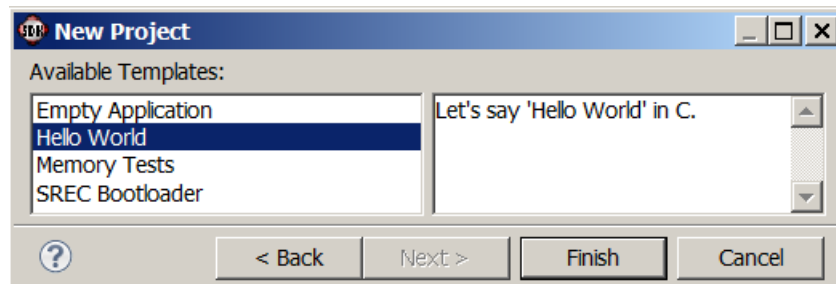


Abbildung 25: Auswahl der Vorlage »Hallo Welt«

»Xio_Out8()« nicht kennt, fehlt entweder die Include-Anweisung für den entsprechenden Header oder der Pfad zum »Board Support Package«. Im letzten Fall empfiehlt es sich, das »Application Project« aus dem Projektbaum zu entfernen, den zugehörigen Verzeichnisbaum zu löschen und das »Application Project« nochmal richtig anzulegen.

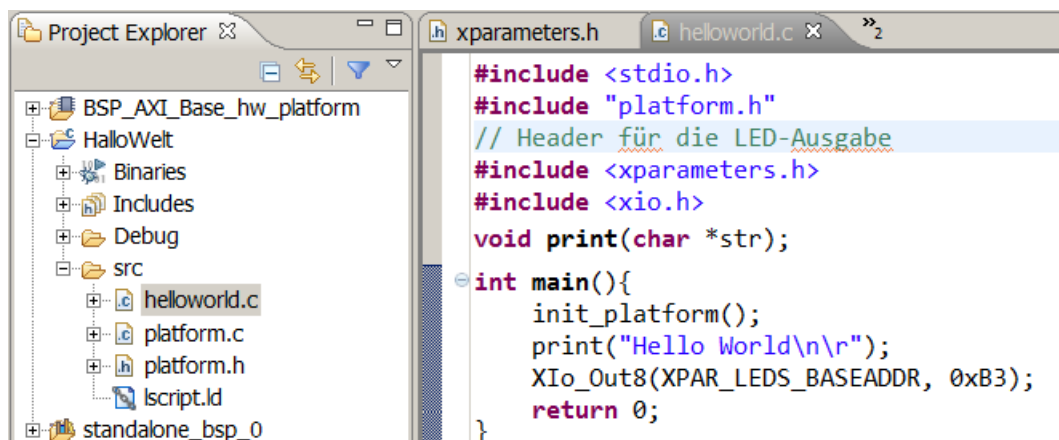


Abbildung 26: Vollständiger Projektbaum und Anwendungsprogramm

Vor dem Start des Programms ist das Bitfile zur Konfiguration der Rechner-Hardware über das Menü

Xilinx Tools ▷ Program FPGA

in den Schaltkreis zu laden. Dabei öffnet sich das Menü in Abb. 27, in dem nur »Program« anzuklicken ist. Falls der Download nicht klappt, ist zu kontrollieren, dass die Baugruppe Strom

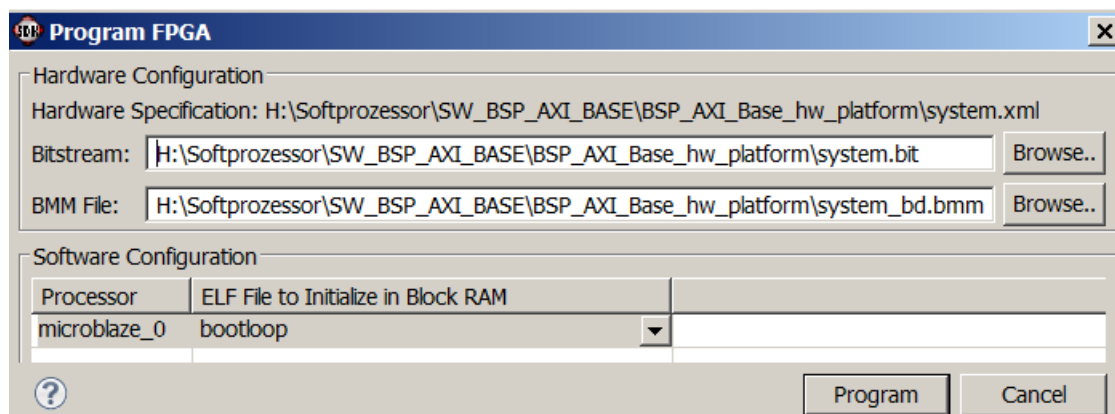


Abbildung 27: Programmieren der FPGA-Hardware

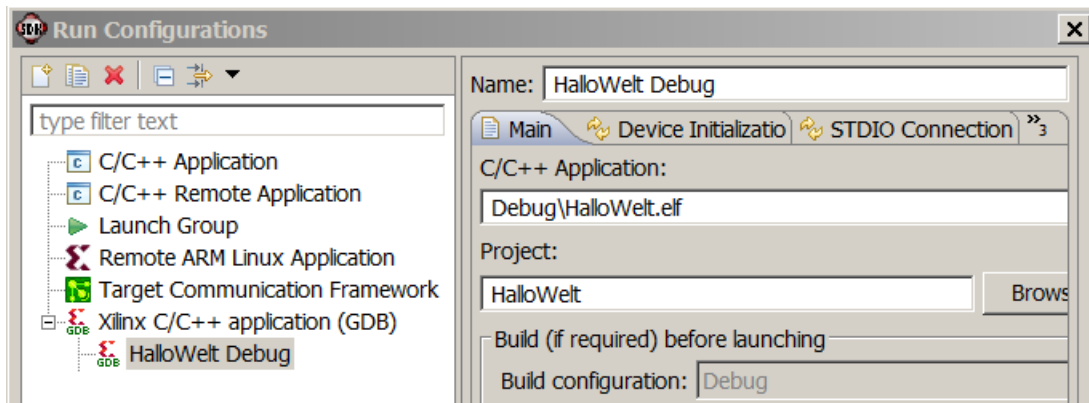


Abbildung 28: Grundeinstellungen der »Run Configuration«

hat, angeschaltet und das Programmierkabel angeschlossen ist. Manchmal helfen auch mehrere Versuche, abziehen und wiederanstecken des Programmier-USB-Kabels und vor allem Geduld. Abschließend ist über das Menü

Run ▷ Run Configurations..▷ »Xilinx c/C++ applicatiion (GDB)

mit einem Rechtsklick auf »Xilinx c/C++ applicatiion (GDB)« eine »Launch Configuration« zu erzeugen. Unter dem Reiter »Main« bekommt diese einen Namen, ein Projekt, ein ausführbares Programm, ... zugeordnet (Abb. 28). In unserem Projekt ist nur unter dem Reiter »STDIO Connection« eine Änderung erforderlich (Abb 29). Hier ist »Connect STDIO to Console« und als Port »JTAG UART« auszuwählen.

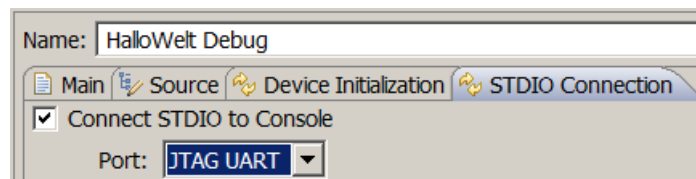



Abbildung 29: Verbindung der Standardausgabe mit der Konsole

Der Programmstart erfolgt mit »Run« bzw. über die Schaltfläche . Nach Laden und Programmstart gibt unser Beispielprogramm auf der Konsole den Text »Hello World« aus (Abb. 30). Das Ausgabebitmuster auf den Leuchtdioden ist »1100 1101« (der bitgespiegelte Wert zu 0xB3).

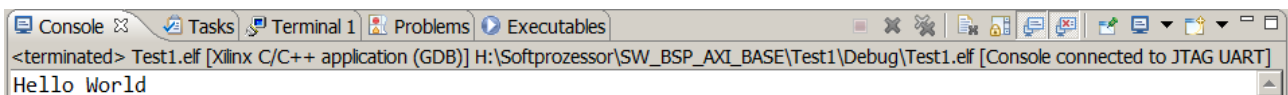


Abbildung 30: Testausgabe auf der Console