



Adım adım FPGA'e MikroBlaze Gömme

Bitirme Öğrencisi:

Cumhur Erdin

040090414

Bitirme Çalışması Danışmanı:

Doç. Dr. S. Berna Örs Yalçın

Adım adım Xilinx 14.1 Microblaze nasıl gömülür?

- Xilinx platform studio (Edk) programı başlatılır. (yönetici olarak başlatılması tercih edilir.)
- Create New Project Using Base System Builder seçilir.
- Project File kısmında browse seçilerek yeni bir klasör oluşturulur ve proje toplu halde durması için herşey içine kaydedilir.
- PLB Systems seçeneği işaretlenir. OK
- I would like to create a new design seçeneği default olarak seçili bırakılır. NEXT
- Board: Kullanılan Xilinx board name ve varsa çeşitli board revisionlarından gerekli olan seçimler yapılır. NEXT
- System: Single Processor System seçilir. NEXT
- Processor: Clock frekansı ayarlanabilir. NEXT
- Peripheral: Kullanılacak dış aygıtlar belirlenir. Microblaze için SDRAM ve dlmb_cntlr dışındakiler kaldırılır. NEXT
- Cache: NEXT
- Summary : Finish
- Hardware-> Create or Import peripheral sekmesinden NEXT
- Create templates for a new peripheral seçilir. NEXT
- To an XPS project seçili bırakılır. NEXT
- İsim verilir. UYARI: verilen isim ve başta seçilen klasör adının aynı olmamasına dikkat edilir. NEXT
- Processor Local Bus (PLB v4.6) seçilir. NEXT
- Include data phase timer seçilmez ve sadece user logic software register seçili bırakılır. NEXT
- Slave interface kısmında bir değişiklik yapılmaz. NEXT
- Kaç adet 32 bitlik register kullanılacağı belirlenir. NEXT
- IP Interconnect NEXT

- Peripheral Simulation Support NEXT
- Peripheral Implementation Support
 - Generate stub 'user_logic' template in Verilog instead of VHDL seçilir. OK
 - Generate template driver files to help you implement software interface seçilir.
- NEXT
- Finish

- Dosya adı -> pcores-> dosya adı->hdl->Verilog-> user logic

- User logic içerisine

- // Implementation başlığının altına çıkış tanımlanır.
 - Örnek: `wire o_wire;`
- // --USER logic implementation added here altına top module çağırılır.
 - Örnek: `andKapi isim (.a(slv_reg0),
 .b(slv_reg1),
 .o(o_wire));`
 - Yukarıdaki o_wire çıkış olarak tanımlanmıştı.

```
3'b001 : slv_ip2bus_data <= {31'd0,o_wire};
```

Yazılan Verilog kodu örnek: Bu kodun bulunduğu V dosyası da user logic'in bulunduğu klasör içine atılır.

```
`timescale 1ns / 1ps
```

```
module andKapi( input a,
                input b,
                output o);
```

```
assign o = a & b;
```

```
endmodule
```

- Hardware-> Create or Import peripheral sekmesinden NEXT

- Import existing peripheral NEXT
- To an XPS project NEXT
- Koyduğumuz isim çağırılır. NEXT YES
- Sadece HDL source files seçilir. NEXT
- Use existing Peripheral Analysis Order file (*.pao) seçilir.
- Browse: pcores içerisinde dosya adı->data-> pao dosyası seçilir. NEXT
- HDL Analysis Information: Add files tan eklediğimiz .v dosyası varsa

Dosya adı ->pcores-> dosya adı->hdl->Verilog-> içerisindeki .v dosyası eklenir. NEXT

- PLBV46 Slave (SPLB) seçilir. NEXT
- SPLB: Port NEXT
- Parameter determine high address kısmına C_HIGHADDR seçilir. NEXT
- Select and configure interrupt(s) seçimi kaldırılır. NEXT
- Parameter Attributes NEXT
- Port Attributes NEXT (çıkış portu kullanmıyoruz bu çalışmada)
- Finish
- Ekrandan Project Local Pcores
 - USER ve altından dosya bus interfaces kısmına taşınır. OK
- PLB kısmından yuvarlağın içi doldurulur ve ya no connection kısmı mb_plb seçilir.
- Adresses kısmından sağ üst köşedeki generate addresses tıklanarak unmapped adresler atanır. (Portlardan dışarı aktarım yapılacaksa port connection kısmı da ayarlanır, ismi kopyanalarak constraint içine yazılır.)
- Project -> Export hardware design to sdk
- Export and Launch SDK (Not: biraz uzun sürebilir.)

SDK

- Dosya konumu OK
- New Xilinx C Project Hello world NEXT FINISH
- Örnek SDK kodu: (hello world c içi)

```
#include <stdio.h>
#include "platform.h"
#include "xparameters.h"
#include "asd.h"

int cikis;

int main()
{
    ASD_mWriteReg(XPAR_ASD_0_BASEADDR, ASD_SLV_REG0_OFFSET, 0x00000001);
    //a giriş
    ASD_mWriteReg(XPAR_ASD_0_BASEADDR, ASD_SLV_REG1_OFFSET, 0x00000000);
    // b giriş
    cikis = ASD_mReadReg(XPAR_ASD_0_BASEADDR, ASD_SLV_REG2_OFFSET); // o
    çıkış

    return 0;
}
```

- Kopyalanması gereken kütüphaneler:
- Dosyaadı->drivers->dosyaadı->src->dosyaadı.h bu dosya
- Dosyaadı->SDK->dosyaadı(hello_world_0)->src->konumuna kopyalanacak
- Dosyaadı->SDK->dosyaadı(hello_world_0)->microblaze_0->include -> konumuna kopyalanacak
- Program FPGA
- Run as hello_world_0.elf
- Gözlemek istediğimiz sonuçlar debug ekranından variables kısmına ekleme yapılarak adım adım gözlenebilir.