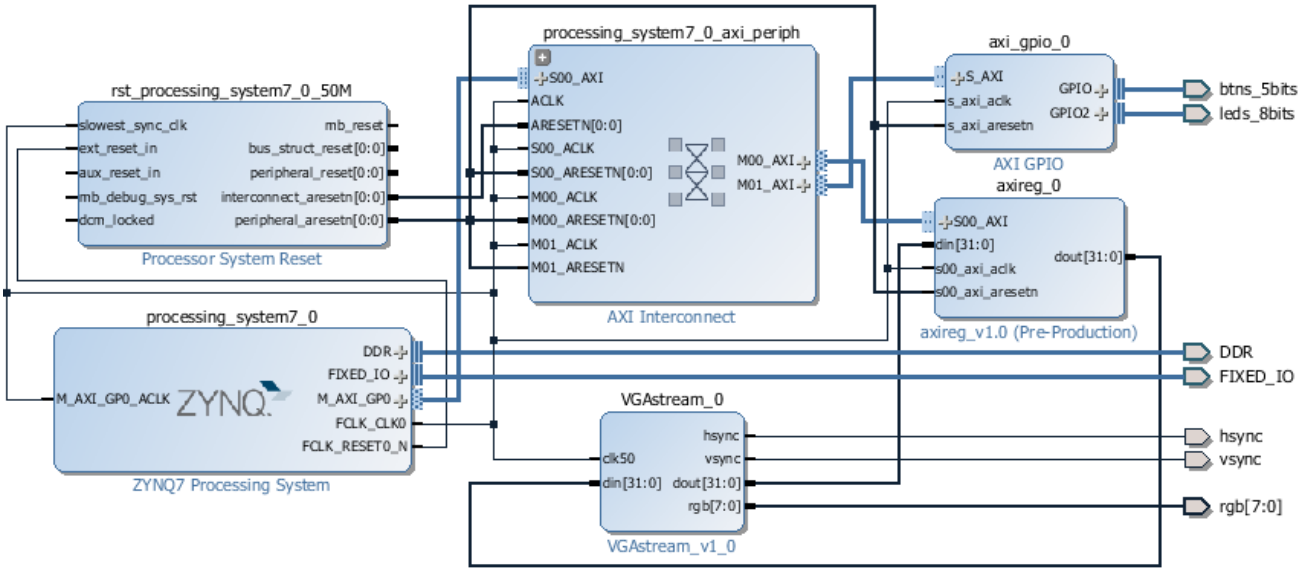
3. vaja: aplikacija

Spoznali bomo:

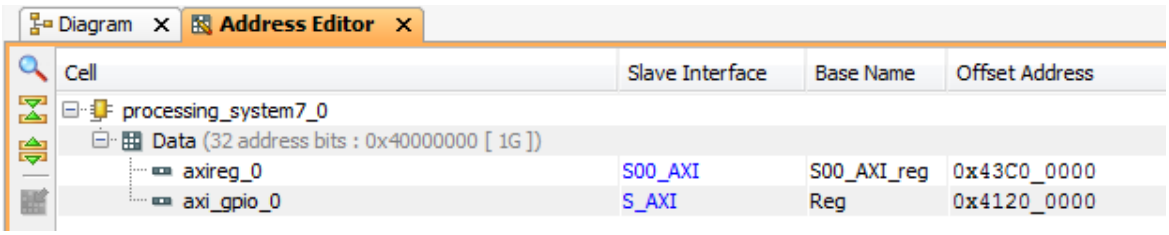
- komponento GPIO (general purpose input-output), kamor bomo priključili LED in tipke
- prevajanje sistema in izdelavo in zagon "hello world" aplikacije
- komunikacijo s periferijo z uporabo gonilnikov
- neposreden dostop do perifernih registrov

3.1 Blokovni diagram z GPIO

- V blokovni diagram dodaj (**Add IP**) knjižnično komponento **AXI GPIO**. Z dvoklikom na komponento nastavi lastnosti ter poveži v sistem (**Run Connection Automation**). Lastnosti:
  - GPIO: btns 5bits
  - GPIO2: leds 8bits



- V blokovnem diagramu sistema se komponentam na vodilu AXI avtomatično dodeli periferni naslovni prostor. Naslove lahko pregledujemo in urejamo v zavihku **Address Editor**. Npr. naslovi registrov komponente za računanje kroga se začnejo pri 0x43C0 0000.

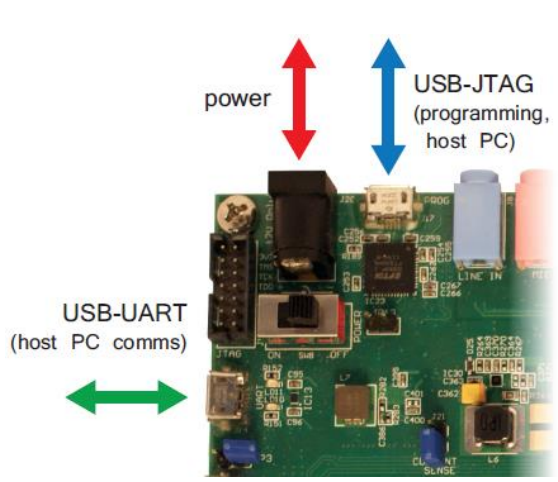


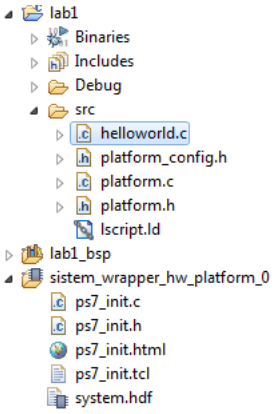
- Pred prevajanjem sistema preverimo, če imamo vključeno datoteko z definicijami priključkov (če tega še nismo storili, vključimo datoteko z **Add Sources**, **Add or Create Constraints**, VGAsistem.xdc)

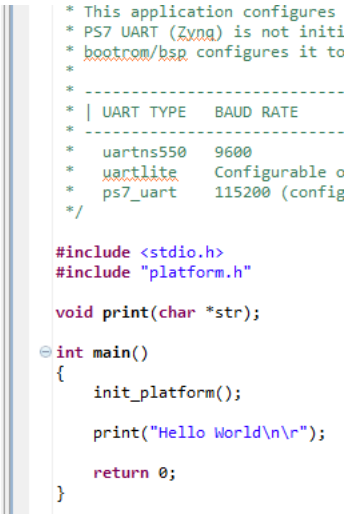
- Naredi HDL model blokovne sheme z desnim klikom na *sistem.bd* in izbiro **Create HDL Wrapper** (izberemo opcijo Let Vivado manage wrapper).
- Prevajanje, sintezo in implementacijo vezja poženi z ukazom **Run Implementation.**
- Ko se odpre okno **Implementation Completed**, izberemo **Generate Bitstream**, kliknemo OK in počakamo da pripravi datoteko za programiranje FPGA.
- Zadnji korak je izvoz datoteke: **File > Export > Export Hardware**, naredimo kljukico na **Include bitstream** in OK.

3.2 Izdelava aplikacije

- Odpri orodje SDK (**File > Launch SDK**) in ustvari novo aplikacijo(**File > New > Application Project**) z imenom lab1, ki ga vnesemo v polje: *Project Name*, v naslednjem oknu pa potrdimo izbiro testne aplikacije z imenom Hello World.
    - V oknu Project Explorer odpri lab1 > src > helloworld.c







```
* This application configures
* PS7 UART (Zynq) is not initi
* bootrom/bsp configures it to
*
* ----------------------------
* | UART TYPE   BAUD RATE
* ----------------------------
*   uartns550   9600
*   uartlite    Configurable o
*   ps7_uart    115200 (config
*/

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

void print(char *str);

int main()
{
    init_platform();

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

    return 0;
}
```

- Priključi ZedBoard na napajanje, JTAG in UART USB konektorja in vklopi napajalno stikalo. Iz menija izberi **Xilinx Tools** > **Program FPGA** in naloži vezje s klikom na **Program**. Na plošči bo po končanem nalaganju zasvetila modra LED.
    - Nastavi serijski terminal: v spodnjem zavihku Terminal klikni na Connect, izberi vrsto povezave: Serial, ustrezna vrata COM in hitrost prenosa 115200.
    - V oknu Project Explorer klikni na aplikacijo (npr. lab1), nato pa jo poženi na razvojni plošči **Run > Run As > 1 Launch on Hardware**. V oknu terminala se mora pojaviti pozdrav: Hello World.

Testno aplikacijo bomo nadgradili in preizkusili komunikacijo s periferijo. Dodajmo knjižnico za dostop do GPIO (xgpio.h), pobrišimo deklaracijo funkcije print in dodajmo nekaj kode v main():

```
#include <stdio.h>
#include "platform.h"
#include "xgpio.h" // gonilnik za GPIO

//void print(char *str);

int main()
{
XGpio gp;
int tipke;

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

XGpio_Initialize(&gp, XPAR_AXI_GPIO_0_DEVICE_ID); // inicializiraj AXI GPIO
while (1) {
 tipke = XGpio_DiscreteRead(&gp, 1);       // beri tipke na 1. kanalu
 XGpio_DiscreteWrite(&gp, 2, 0xF0+tipke);  // nastavi LED na 2. kanalu
}
```

- Program se avtomatsko prevede, ko ga shranimo. Poženi program na razvojni plošči in preizkusi delovanje tipk in LED.

- Dodaj stavke za komunikacijo z registri in poskusi narisati in s tipkami premikati krog po zaslonu.

```
#include "xil_io.h" // neposreden dostop do registrov
#include "sleep.h"  // knjižnica z  uporabnimi funkcijami za zakasnitve

  Xil_Out8(0x43c00000, 30); // dostop do prvega registra
```

Poročilo

Napiši kratko poročilo, ki naj vsebuje:

- opis komponente AXI_krog in izsek iz VHDL kode, kjer se izvaja proces izračuna enačbe kroga in izhoda dout.

- opis testne strukture in predstavitev simulacije delovanja komponente

- predstavitev testnega sistema z VGA izhodom s kratko razlago in opis zasedenosti programirljivega vezja po prevajanju sistema.
    - Podrobnejše podatke o zasedenosti vezja dobimo v zavihku Reports orodja Vivado. Iz poročila postopka razmeščanja elementov (Place Design) odčitamo % zasedenosti FPGA rezin (Slice), število registrov (flip-flopov)…

- izsek iz programa v jeziku C s katerim ste preverili delovanje na razvojnem sistemu in kratka razlaga.