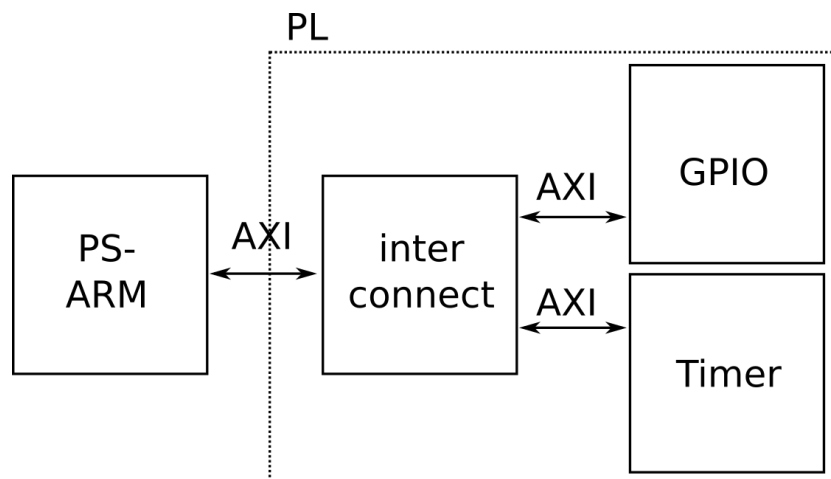


## 3.2 Имплементација хардвера - Vivado IP Integrator

Пројектовани хардверски систем може се имплементирати у многим алатима различитих фирми. Избор алата за имплементацију зависи од циљане технологије. У случају да се коначна имплементација спроводи на Field Programmable Gate Array (FPGA) чипу фирме Xilinx за имплементацију хардверског система може се користити Vivado IP Integrator алат. Примена овог алата биће укратко улустрована на имплементацији једноставног система (слика 3.7), чија виртуелна платформа је већ развијена (слика 2.22). Имплементација хардвера може се посматрати и као мапирање виртуелна платформе на платформу намењену имплементацији. Основна разлика између ова два система је што ће се уместо TLM магистрала користити Advanced Extensible Interface (AXI) магистрале.



Слика 3.7: Систем за имплементацију

FPGA чип садржи два главна дела: Processor System (PS) и Programmable Logic (PL). PS део чипа је мултипроцесорски систем који садржи већи број Advanced RISC Machine (ARM) процесора. Овај део система служи да извршава развијени софтвер. Хардверске компоненте имплементирају се унутар PL дела чипа.

### 3.2.1 Прављење пројекта

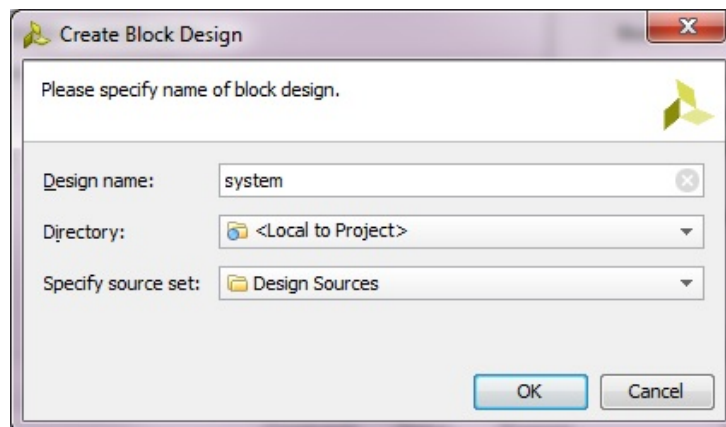
Први корак у имплементацији система је прављење пројекта.

1. Урадити **File -> New Project** да би се отворио дијалог за прављење пројекта.
2. Потребно је специфицирати да је у питању RTL пројекат и не треба специфицирати изворне датотеке. Кликнути **Next**
3. Потребно је специфицирати конкретан FPGA чип који ће се користити за имплементацију или је потребно специфицирати развојну плочу. Подразумева се да ће приказани пример бити имплементиран на Zybo плочи, па је њу потребно одабрати. Кликнути **Finish**.

### 3.2.2 Прављење блок дизајна

Унутар Vivado алата, за пројектовање система користи се стандардизована Extensible Markup Language (XML) шема за опис система који се састоје од IP блокова: SPIRIT's IP-XACT (IP-XACT). У оквиру Vivado алата IP-XACT датотеке се сачувавају са екстензијом \*.bd. Иако су XML текстуалне датотеке, ретко се обрађују помоћу уређивача текста. Због свог формата, погодне су за рачунарску обраду. У оквиру Vivado алата, ове датотеке обрађују се графички.

1. Урадити **Create Block Design** унутар **Project Manager**. Овим се прави IP-XACT датотека.
2. Потребно је дати име систему који се прави, на пример **system**.



Слика 3.8: Блок дизајн

3. У оквиру Vivado алата појавио се празан поглед за прављење блок дијаграма. У оквиру овог погледа могуће је додавати и спајати хардверске IP блокове.
4. Први блок који ће бити додат у блок дизајн је Zynq7 PS блок. Овај IP блок служи да се ARM процесори конигуришу за коришћење у оквиру система. Да би се додао IP блок потребно је отворити одговарајућ дијалог: десни клик на дијаграм поглед, па Add IP. . . . Потом у Search откуцати Zynq и одабрати једини преостали IP. Након овога у дијаграм погле убачен је IP блок (слика 3.9).



Слика 3.9: Zynq IP

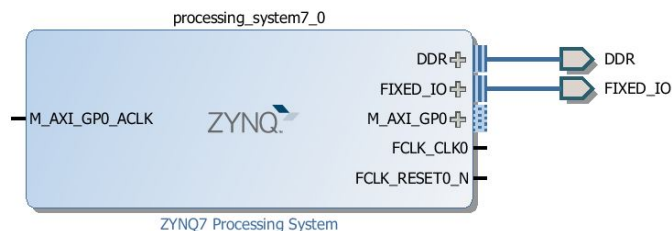
### 3.2.3 Конфигурисање Zynq PS IP блока

Zynq PS IP омогућава конфигурисање свих делова процесорског система унутар чипа. Овај систем обухвата ARM процесорске корове али и већи број других блокова, од којих су неки: Double Data Rate (DDR) меморијски контролер, контролер прекида, контролери разних улазно излазних протокола.

Zynq PS IP садржи већи број подразумеваних приступа (слика 3.9) Приступу DDR служи за комуникацију са DDR меморијом. Приступу FIXED\_IO служи за комуникацију FPGA чипа са спољним компонентама. Овај приступ је од значаја зато што садржи Universal Asynchronous Receiver-Transmitter (UART) комуникацију помоћу које ће FPGA чип моћи да шаље поруке рачунару на коме се развија систем. Приступ M\_AXI\_GPO је AXI магистрала помоћу које PS део чипа комуницира са хардверским компонентама у PL делу чипа. Синхронизациони сигнал за ову магистралу везан је на улаз M\_AXI\_GPO\_ACLK. Излазни синхронизациони сигнал намењен PL делу чипа је FCLK\_CLK0. Излазни ресет сигнал је FCLK\_RESET0\_N.

### 3.2.4 Повезивање са спољним компонентама

Да би се PS део повезао са спољним компонентама потребно је DDR и FIXED\_IO приступе Zynq PS IP блока прогласити излазним приступима целог система. Додано, потребно је помоћу XDC датотеке ограничити на које пинове FPGA чипа се везују појединачни сигнали ових приступа. Та процедура може бити дугачка и приметна. Vivado алат садржи информације са којом плочом ради и стога, овај задатак је у потпуности аутоматизован. Све што је потребно урадити јесте активирати Run Block Automation у оквиру дијаграм погледа. Све вредности у оквиру дијалога који се добије потребно је оставити непромењеним (слика 3.10).



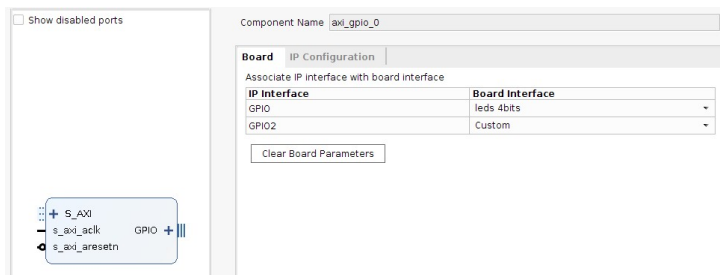
Слика 3.10: Спољне везе система

Након што је покренута наведена аутоматизација може се видети да су спољни приступи означени као излази целог система.

### 3.2.5 Додавање IP језгара

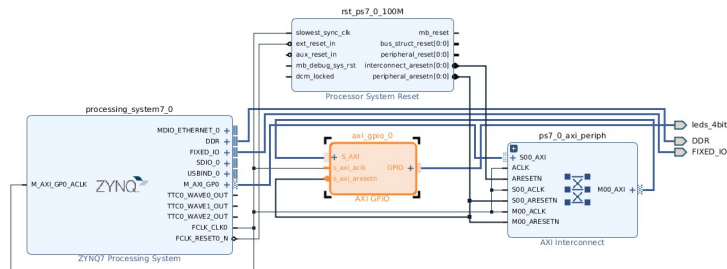
GPIO језгро и језгро за мерење времена постоје као већ готове компоненте у оквиру каталога IP компоненти фирме Xilinx.

1. Урадити Add IP... и додати AXI GPIO IP.
2. Конфигурисати језгро (двоструки клик на језгро у блок дијаграму), тако да излази овог блока буду повезани са Light-Emitting Diode (LED) на Zybo плочи: Board таб -> GPIO -> Board Interface -> leds 4bits (слика 3.11). Пошто алат има информације са којом плочом ради, након овог подешавања, аутоматски ће се подесити ширине излаза и биће додата ограничења која ће повезати излаз овог блока са LED.
3. На располагању је нова аутоматизација коју се започиње са Run Block Automation. Потом треба одабрати опцију All Automation и затим се аутоматизација покреће са ОК. Ова аутоматизација



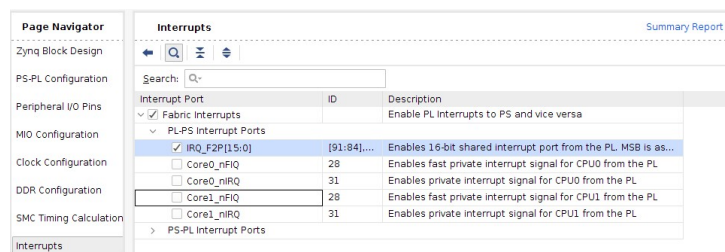
Слика 3.11: Конфигурација GPIO

убациће интерконект компоненту и помоћу ње повезаће GPIO IP са процесором. Додатно убадиће се IP блок који служи да се ресетују све периферије унутар система (слика 3.12).



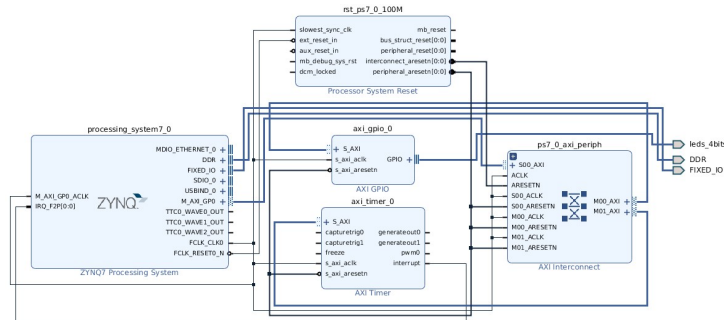
Слика 3.12: Систем са GPIO

4. Додати AXI Timer језгро у систем, па затим покренути аутоматизацију која се појави.
5. Да би се омогућили прекиди са AXI Timer језгра, потребно је конфигурирати PS да може да прима прекиде са PL дела чипа. Потребно је конфигурирати Zynq PS IP -> Interrupts и одабрати Fabric Interrupts и IRQ\_F2P[15:0] (слика 3.13).



Слика 3.13: Додавање прекида

6. Повезати излаз AXI Timer interrupt са улазом Zynq PS IRQ\_F2P. Након овог повезивања добија се коначна хардверска имплементација система (слика 3.14).



Слика 3.14: Имплементирани систем

Аутоматизације служе да олакшају повезивање система. Све кораке које аутоматизација уради могуће је постићи и коришћењем појединачних команди. На пример, могуће је убацити интерконект као и сваки други IP блок и потом је подесити по потреби. Аутоматизација олакшава неке кораке али може да уради и нешто што дизајнер не жели. Ову могућност треба користити опрезно и потребно је бити свестан свих подешавања које аутоматизација ради.

### 3.2.6 Адресна мапа

По потреби адресе периферија у систему могу се мењати у табу Address Editor (слика 3.15). Свако интерконект језгро у себи садржи табелу на основу које може да успешно рутира AXI пакете унутар система.

Cell	Slave Interface	Base Name	Offset Address	Range	High Address
processing_system7_0					
Data (32 address bits : 0x40000000 [ 1G ])					
axi_gpio_0	S_AXI	Reg	0x4120_0000	64K	0x4120_FFFF
axi_timer_0	S_AXI	Reg	0x4280_0000	64K	0x4280_FFFF

Слика 3.15: Адресна мапа

### 3.2.7 Валидација блок дијаграма

Читав блок дизајн представља интеграцију постојећих IP блокова у систем. Опис овог система чува се у IP-ХАСТ датотеци. Vivado Integrator алат омогућава да се неке грешке у систему уклоне само на основу овог описа. На пример, две AXI магистрале подешене да имају различите ширине могу бити спојене у IP-ХАСТ опису. Овакве грешке могу се уочити коришћењем алата за валидацију блок дизајна. Потребно је покренути **Tools -> Validate Design**, након чега се добија извештај са могућим проблемима који постоје у опису. Уколико постоје грешке потребно их је уклонити па тек онда прећи на наредне кораке.

### 3.2.8 Синтеза и имплементација

Vivado алат захтева да дизајн намењен синтези буде у HDL формату. Другим речима, није могуће покренути синтезу над IP-ХАСТ датотеком. У оквиру Vivado алата постоји могућност да се аутоматски направи HDL датотека која ће представљати систем описан у IP-ХАСТ формату. Ова датотека се прави тако што се одабере IP-ХАСТ датотека у **Source** таби, десни клик, па се покрене акција **Create HDL Wrapper**. Затим се добија дијалог у ком се бира да ли да се добијена датотека аутоматски ажурира када се промени блок дијаграм или датотеку треба да мења дизајнер. Оба приступа имају своје предности. У овом примеру пустиће се да алат све аутоматски ажурира. Након ове акције добија се HDL датотека која се посматра као модул који је намењен синтези.

Након припреме, може се покренути синтеза а затим и имплементација система. Потребно је покренути акцију **Generate Bitstream**. Имплементација система може да потраје дуже времена у зависности од комплексности система. Ово време је значајно дуже од времена потребног да се инкрементално компајлира софтвер и може трајати од неколико минута па до неколико дана. Када се имплементација заврши добија се бинарна датотека са екстензијом **\*.bit** помоћу које се конфигурише хардверски део FPGA чипа.

### 3.2.9 Извоз хардверског описа

Развој софтверских компоненти Zynq платформе одвија се у другим алатима. Софтверским алатима је потребан опис хардверских компоненти система. Да би се добио овај опис, потребно је након имплементације покренути **File -> Export -> Export Hardware**. У дијалогу који се појави потребно је селектовати **Include Bitstream** и

потом покренути акцију. Нако ове акције, биће направљен посебан директоријум намењен развоју софтвера у оквиру кога ће се налазити и опис хардверског система. Директоријум ће имати исто име као и Vivado пројекат, само ће му бити додат наставак `*.sdk`. У оквиру тог директоријума, налазиће се датотека `system_wrapper.hdf` која је добијена извозом хардвера.

Један од алата за развој софтвера намењеног Zynq платформи је Xilinx Software Development Kit (SDK). Овај алат могуће је покренути из Vivado алата након имплементације акцијом: **File -> Launch SDK**. Ова акција покренуће SDK алат, који ће аутоматски учитати извезени хардверски опис, нако чега је могуће почети развој софтвера са имплементирану хардверску платформу.