

於ARM-SoC平台實作與驗證16位元的RISC

Implementation and verification 16-bit of RISC on ARM-SOC platform

指導老師：林寬仁 副教授

學生：邱錦洋、林敬昆

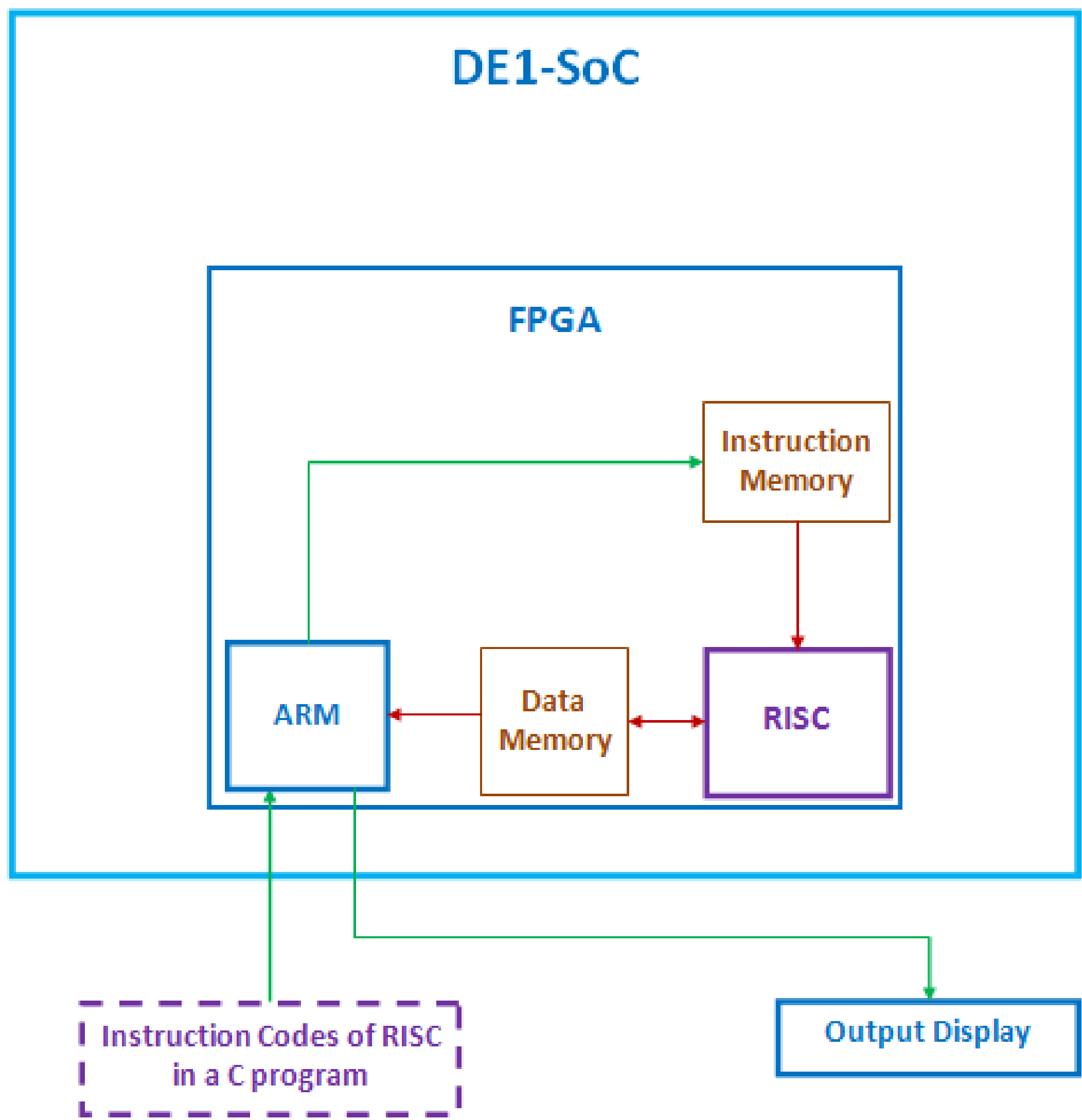
輔仁大學 電機工程學系 大學部專題生

摘要

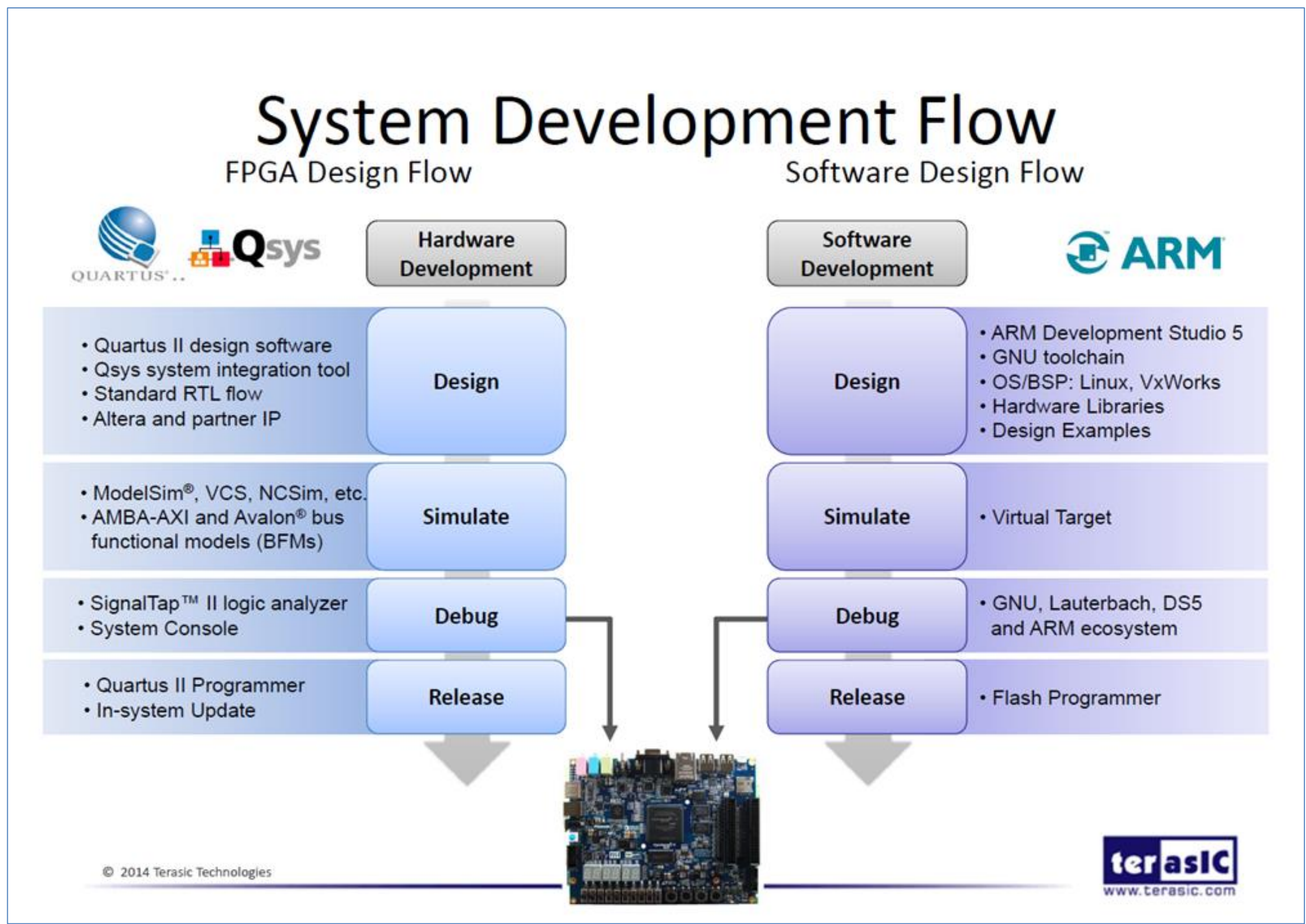
精簡指令集(Reduced Instruction Set Computing)，又稱RISC，屬於電腦中央處理器(CPU)的設計模式。RISC以許多基礎指令來取代一個複雜指令，儘管程式碼變得較多，但卻能讓處理器簡化，進而提升效能。由於指令長度的固定，方便CPU進行解碼，使其實作更加的容易。

本專題開發硬體及程式，使用Quartus II 14.1、Qsys、GNU C等工具。此RISC可擴充為一個特定運用之處理器，如加解密或影像處理器，亦能做為ARM之協同運算處理器。

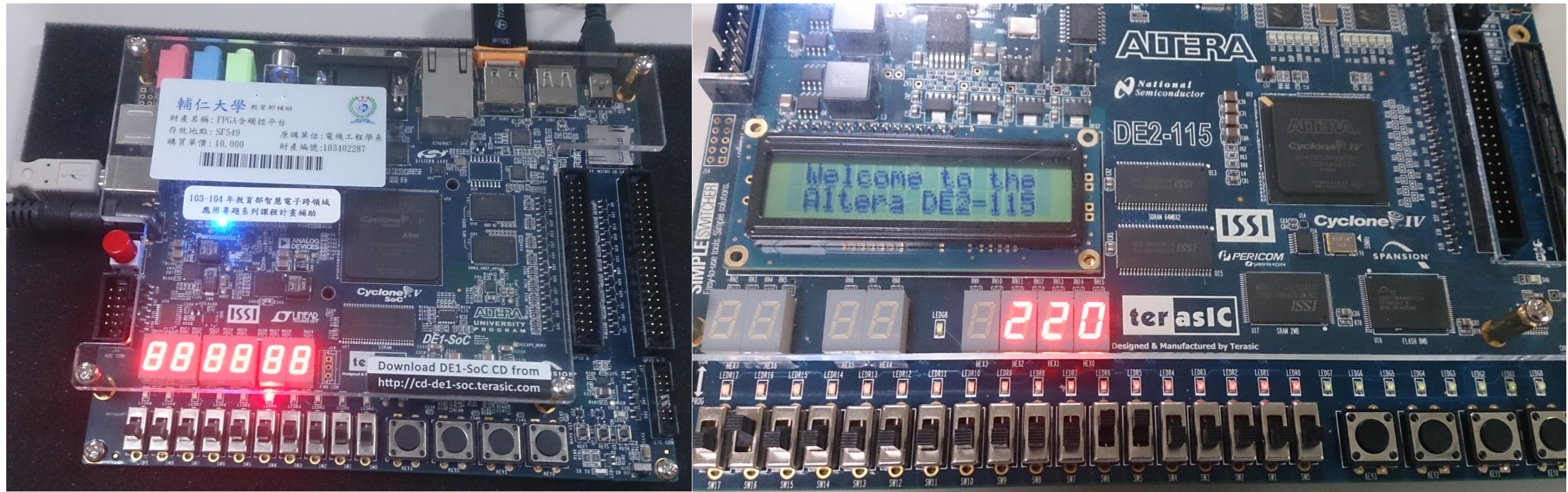
系統示意



此RISC將在內嵌ARM之FPGA上進行(DE1-SoC)，在支援ARM的程式開發環境下，透過C Language撰寫的程式，將指令載入至記憶體，接著啟動RISC執行，驗證其功能。其執行結果透過ARM執行之程式可顯示在螢幕上或開發平台之LCD/LEDs。



開發流程

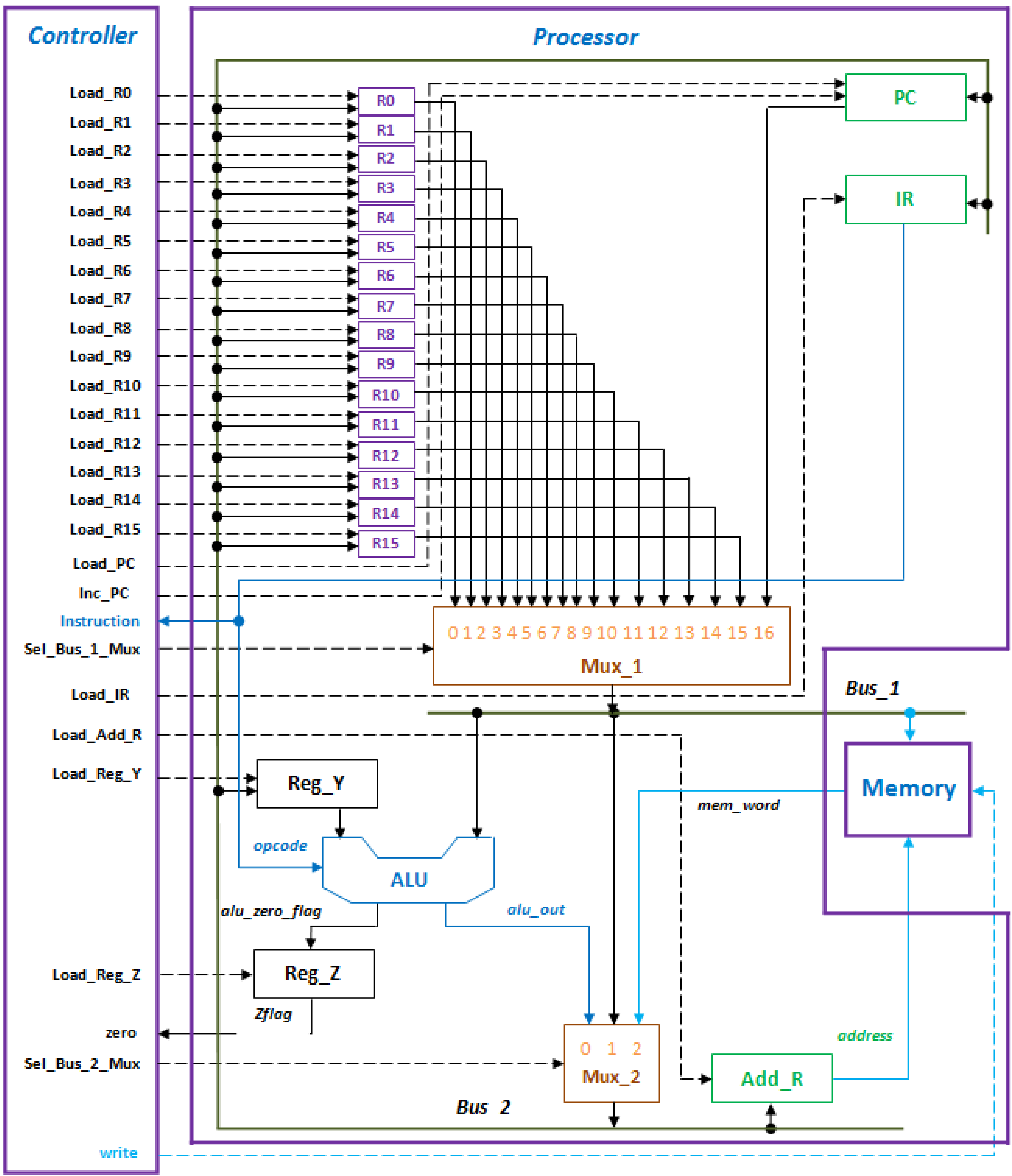


(DE1-SoC)(內嵌ARM)

顯示至開發平台的LED

(DE2-115)

RISC架構



本專題參考Advanced Digital Design with the Verilog之RISC_SPM電路，將原本的8位元擴充至16位元，並設計新指令，添加與ARM溝通介面電路，而以Verilog Code實現之。

指令

每個指令由Operation Code(8-bit)、Source(4-bit)、Destination(4-bit)組成，Operation Code決定做的事；Source、Destination，分別代表兩個不同的暫存器。

Operation Code								Source				Destination			
0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1
SUB								R2暫存器				R1暫存器			

以組合語言的寫法類似於：SUB R1, R2
實際上會執行的動作：Destination - Source => Destination
實際上會執行的動作：R1 - R2 => R1 (將R1-R2的值存回R1)

指令名	指令碼			執行動作
	Opcode	Src	Dest	
NOP	00000000	×	×	無
ADD	00000001	Src	Dest	Dest + Src => Dest
SUB	00000010	Src	Dest	Dest - Src => Dest
AND	00000011	Src	Dest	Dest & Src => Dest
NOT	00000100	Src	Dest	~Src => Dest
RD	00000101	Ram 編號	Dest	Ram(Src)[NextData] => Dest
WR	00000110	Src	Ram 編號	Src => Ram(Dest)[NextData]
BR	00000111	×	×	Rom[NextData] => PC
BRZ	00001000	×	×	Rom[NextData] => PC(If Zero)
MUL	00001001	Src	Dest	Dest[7:0] × Src[7:0] => Dest
ONE	00001010	Src	Dest	計數 Src 有幾個「1」=> Dest
LRT	00001011	Src	Dest	Dest 左旋 Src 位元 => Dest
CTR	00001100	×	Dest	NextData => Dest
LFSR	00001101	Src	Dest	LFSR Dest 於 Src 位元 => Dest
STP	00001110	×	×	使 RISC 停止讀取指令

RISC的指令表



2015 輔仁大學電機工程學系
大學部專題成果展

