第二章 文獻探討

第一節 大專院校電機電子群專業能力分析探討

在台灣，四年制技術學院或科技大學下承高職，二年制技術學院或科技大學則銜接二專、五專，按照預定培育專業目標、能力歸類結果及規劃原則，除培養電機電子業之高級工程技術服務人員外，亦應以技術學院或科技大學為基礎，銜接研究所碩士班繼續進修，以進一步培養高級科技人才[12]。因此在技專院校之專業科目開課原則，應以專業分工、就業實務導向、先廣後專等原則來規劃，盡量貼近未來產業發展的脈動。

技術學院與科技大學之電機電子類群，以培養電機電子業之高級技術人才為目標，其從業人員可涵蓋電機、電子、資訊、電信、控制、光電、冷凍空調等範疇，除應具備大學教育應培育之通識能力外，並應達成下列專業能力目標[39]：

一、培養應用、檢測、分析、設計、管理、研發電機電子相關設備、產品之實務技術與知識之能力。

二、培養兼具人文素養與科技應用，及適應環境變遷之多元能力。

三、培養敬業、負責、勤奮及團隊合作之服務態度與職業道德。

四、培養繼續進修之興趣與能力，奠定終身學習及生涯發展之基礎。

另外，在劉世勳的研究中亦指出，技術學院或科技大學階段，所培養之高級技術人員，其預定培育之專業能力以設計、管理、研發、應用、檢測、分析電機電子設備產品為主要目標。並將專業能力規劃為共同核心能力及群組能力二大項，共同核心能力為電機子高級技術人員共同必須具備之基礎能力，群組能力則是專業分工之後可選擇準備之專業能力。二者皆以必備能力、應備能力及外圍能力。在共同核心能力方面能力如下幾點：

(一)必備能力

1. 應用、分析數位邏輯電路之能力。
2. 應用、檢測、分析、設計電子電路之能力。

3. 應用、分析電腦軟體之能力。
4. 應用、檢測、分析網際網路之能力。

(二)應備能力

1. 應用電機電子工具儀器之能力。
2. 設計、研發數位邏輯電路之能力。
3. 研發電子電路之能力。
4. 設計電腦軟體之能力。
5. 設計、管理、研發網際網路之能力。

(三)外圍能力

1. 分析、設計電機電子工具儀器之能力。
2. 操作、裝置、管理、研發電腦軟體之能力。

在必備能力其中二項為:「應用、分析數位邏輯電路之能力」及「應用、檢測、分析、設計電子電路之能力」。在應備能力則有:「設計、研發數位邏輯之能力」和「研發電子電路之能力」[39]。

因此，由以上的應具之專業能力分析在技專校院、專科大學層級所培養之高級技術人員專業能力部分,「設計及研發電子電路能力」是不可或缺之一環。故而在有限的時間內，考量專業能力的訓練與培養，課程規劃與教學設計亟需被重視。

第二節 HDL 硬體描述語言

壹、硬體描述語言

PLD 已是設計者最常使用的一種 IC 設計，這一類的產品提供公司，如:Xilinx、Altera、Atmel、Lattice…等等,提供包含電路輸入(Design Entry)、編譯等處理、驗證、模擬、及晶片規劃與電路板測試等等方便的環境，所以整個設計流程的總和時間相當的短,以便能爭取快速上市的商機(Time to Market) [31]。

一般的 PLD 若使用人工去規劃內部邏輯陣列,如果電路複雜且龐大，幾乎是不可能達成的事，因此元件製造商通常都會提供一些 CAD 工具，

或是找相關的軟體廠商合作，提供另一種設計的方式，稱 3rd party development tool，如 VHDL、Verilog、OrCAD、Synario、…等等，以方便設計工作的進行，並提供標準工業網路列印檔輸入(Industry-Standard CAE Entry, EDIF)，可與其他的輔助設計軟體互相溝通。

使用 HDL 來進行硬體設計，是近年來在積體電路設計領域中新興的一股潮流，愈來愈多的人對這個題目感到興趣，並致力於這方面的研究。其中最為人注目的二個語言，其一是 Verilog 的 HDL，其二是本研究所要介紹的 VHDL。前者是廣為業者所使用的邏輯模擬器中的語言，後者則是業最早獲得 IEEE 協會認可的業界標準，兩者各有其擁護者，但是發展目標一致，都是希望能協助設計者縮短積體電路的設計流程。

Palnitlear[23]與 Sagahyroon[1]提到，與用繪製電路圖設計方式相比，HDLs 提供設計者以下的優點：

一、設計能被轉換成各種不同層級的抽象層次，設計者能將其設計寫為暫存器轉換(Register Transfer Level, RTL)層級的描述。邏輯合成工具能自動地將設計轉換成各種可能的架構。

二、用 HDLs 來描述其設計，能提早在設計周期期間做功能上的驗證。因為設計者是於 RTL 層級，能有效的更改其 RTL 描述，直到完成所要的功能。大多數的設計缺失都能在此時解決，這將會顯著地減少設計的時間。

三、以 HDLs 設計類似於計算機程式設計，文字化的描述和註釋易於發展及偵錯電路。用邏輯閘層級(Gate-Level)繪圖方式是難於設計非常複雜的數位電路。

貳、選用 VHDL 的原因

VHDL 基本上是美國國防部在 1970 末期到 1980 初一個名叫 VHSIC 計劃（Very High Speed Integrated Circuit）的附屬產物。這個計劃的目的在生產下一代的積體電路，計劃參與者嘗試將設計到製造各個階段的技術推展到極致。在進行這個計劃的初期，設計者以邏輯閘層級的軟體工具來進行 IC 的設計工作，結果產生數以百計千計的邏輯閘設計，當時的軟體設

計工具無法有效率的處理如此巨大複雜的線路，於是設計者開始思考一種新的設計方式。1981 年一種稱為 VHSIC Hardware Description Language(VHDL)的硬體描述語言被提出來，它的主要目的有二，其一是用來描述複雜的電路設計；其二是希望建立一個業界共通的標準，使所有使用 VHDL 語言設計線路的設計者可以透過標準格式來分享彼此的設計經驗。1986 年 VHDL 在 IEEE 協會中正式被提出，經過一連串的審查和修改，終於在 1987 年 12 月成為 IEEE1076 標準。自此之後，VHDL 廣為許多電腦輔助設計公司、大學院校等所採用，直至今日 VHDL 更支援絕大多數的自動化設計工具，並廣為應用至設計週期間的模擬、合成及測試階段。

Sagahyroon[1]認為，在此強大的趨勢下，VHDL 在 IC 設計週期及設計自動化的前題下，大專院校應加重提供以 HDLs 為基礎的設計課程供學生學習應用，並盡可能提供更多 VHDL 語言設計的相關課程。

標準化的硬體描述語言可以使電路設計更具有可攜性。可攜性的電路設計語言，可以適用於不同的邏輯元件或製程技術，亦可適用於不同的設計系統。而 VHDL 除了具有可攜性的優點外，還具有下列之優點[34]：

一、功能涵蓋廣：VHDL 自 ASIC 設計到電路板設計或大型系統的設計均可使用。

二、設計彈性化：VHDL 電路描述語言是以行為化描述方式來設計，所以具有設計快速、更改維護容易及方便除錯的特性。一個原始的 VHDL 程式，可以透過不同的編譯器或合成器加以合成，再依不同的邏輯元件及製程技術，將電路實現。

三、不同的描述風格：VHDL 提供了許多不同的描述風格，來適應大小、複雜性不同的電路設計。一般提供的描述風格有連線關係、順序性敘述、共時性敘述及布林方程式等。

再者，依據 Douglas J. Smith 1996 年的資料顯示[15]，如圖 2-1 所示。VHDL 在較大的系統化設計中，提供比 Verilog 更佳的行為化設計層次。由以上種種條件分析之下，因此選用 VHDL 為本研究之主要 HDLs 設計程式語言。

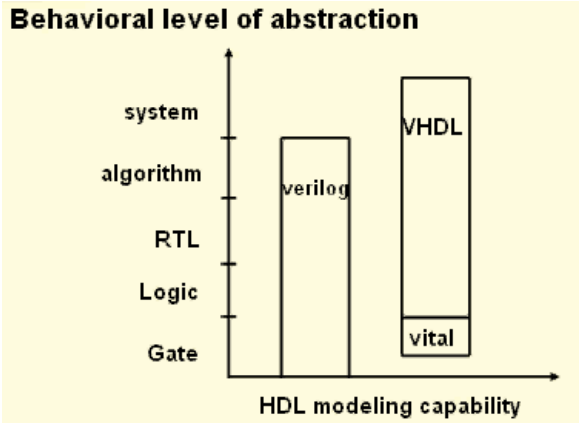

圖 2-1 VHDL 與 Verilog 行為化描述能力圖

第三節 計算機組織與設計教學相關文獻

台灣 IC 設計業界在電腦及消費性 IC 設計的表現上相當耀眼，但在通訊方面則頗有力不從心之感。目前在網路和通訊高階系統領域中，整合影像、資料數據和語音的數位匯聚潮流正持續進行中，而由 FPGA 晶片所提供的快速及具彈性的研發模式，可幫助台灣科技產業影像整合、數位通訊等等高階數位領域。設計者在設計 CPU 時不見得可以想到所有的應用，但針對不同的使用者需求，此類產品能配備不同的功能，因此保留設計彈性是相當必需的。

Harrison 及 Jones 認為，應用 FPGA 晶片於計算機系統相關科目教學，有以下幾個目標[5]：

一、因應市場技術人才需求：以人力市場供需的原則來看，產業界所採用的生產技術人員，應由學校及產業共同培植。

二、設備及圖書更新之問題：傳統的實習課程及教材，因產業技術變動迅速，學校在設備及圖書的更新方面，容易因經費、政策、經營者…等等的問題，不容易與產業同步。

三、培育高品質的技術工程師：每個教育階段的教學目標不同，為培育高品質的技術工程師，應用配合產業的生產技術，更新課程、實驗及教學，是不容置疑的。

同時，Harrison 及 Jones[5]更提到，應用 FPGA 於計算機系統相關科目

教學，其主要目標及次要目標，如下幾點所述：

一、主要目標：(1)增進企劃及建置複雜的電子系統設計能力。

(2)加強群組工作、溝通技能。

二、次要目標：(1)增加學生使用 CAD 工具的機會。

(2)提升學生個人專題設計執行之技術能力。

為因應產業界技術進步的腳步，學校相關於計算機組織與設計的教學與實作，亦應跟隨產業脈動進行調整。因此，採用與產業界相似的產品開發流程，降低學校與產業間的差距，提升學生畢業後隨即就業的能力。國內外相關於計算機組織設計教學的文獻眾多，在課程計劃中有限的時數安排之下，教學者如何能讓學生懂得模組化的設計觀念並導入整體設計架構概念，在此提出幾篇發表在期刊中的文章，探討其特點，以為本論文之參考[11][24][21][20]18]。

壹、Ochi

H. Ochi 在 1997 年發表的「ASAver.1: An FPGA-Based Education Board for Computer Architecture/System Design.」[11]中提出，在大學有限的時間中提出一套可行的教學計畫，使得大學生能了解並設計發展 16 Bits Pipelined RISC microprocessors，同時設計一全新的教育板(Education Board)，能將電腦及硬體系統教學連貫，成為學生自身的學習經驗。

為使每一個大學生盡可能利用課程中有限的時間，達成自行發展設計一「16 Bits 管線式 RISC 微處理器」的相關經驗，而提出以 FPGA 為基礎的計算機系統設計教學計畫，且由教師自行設計一適用於此教學計畫的實驗平台。而使用 FPGA 進行教學，在近幾年來受教育者重視的原因有以下三點[11]：

一、以一個 16 Bits 複雜的微處理機系統，能輕易的在單一個 FPGA 裝置上實現，不需額外大量使用邏輯閘 IC 及複雜的接線過程。

二、FPGA 可以快速地被重新規劃，在適當使用之下，一顆 FPGA 晶片即能進行多次的教學與實驗，且在設計過程能較容易進行偵錯(debug)工作。

三、以 FPGA 為基礎的教學需求花費很低，除了在 CAD 工具上的 license 申請可能需付費之外，一台個人電腦、一套 CAD 工具，在設計完成後，即能在個人電腦上進行軟體功能模擬工作。

在實施方式方面，為使能將發展 Pipelined RISC processor 的實驗融入在有限時間的課程中，Ochi 建議一種新的方式，按四個步驟的順序進行，分別為：

一、在 FPGA 上設計及實現一個簡單的 Pipelined(1-CPI) RISC Processor，由小模組的設計與實現為開始，漸漸導入整體架構的設計與實作。

二、發展 Processor 設計並修正設計期間所發現的缺失。仔細觀察 Processor 在執行時的運作特性。

三、使用檢查程式評估 Processor 是否發展成熟，並嘗試挑戰減少指令的執行週期的相關進階設計。

四、學生進行總結報告，並歸納設計結論。

Ochi 在課程實施中提供簡單的 16 Bits Pipelined CPU 結構做為教學及實驗的依據，並將其命名為 ASAP-0/f0、ASAP-0/f1、ASAP-0/f2。此 CPU 僅建立 10 個指令集，並採用 Verilog 來撰寫。指令集如表 2-1 所示。

表 2-1 Ochi ASAP-0/f0 實作指令表

Mnemonic	(MSB) Instruction Code (LSB)				Verilog-HDL Pseudocode

LD Ra, d(Rb)	00	Ra	Rb	d	// LoaD from memory
					gr[Ra] <= mem[gr[Rb]+d];
					pc <= pc+1;

ST Ra, d(Rb)	01	Ra	Rb	d	// STore to memory
					mem[gr[Rb]+d] <= gr[Ra];
					pc <= pc+1;

LI Ra, d	10	001	Rb	d	// Load Immediate
					gr[Rb] <= d;
					pc <= pc+1;

LHI Ra, d	10	011	Rb	d	// Load High Immediate
					gr[Rb] <= d<<8 | gr[Rb]:255;
					pc <= pc+1;

B Ra, d	10	111	000	d	// Branch pc <= pc+1+d;
BNE d	10	111	100	d	//Branch Not Equal pc <= (!z) ? pc+1+d : pc+1;

ADD Ra, Rs	11	Rs	Rd	0000 0000	// ADD arithmetic
					gr[Rd] <= gr[Rd] + gr[Rs]
					z <= ~ |(gr[Rd]+gr[Rs]);
					pc <= pc+1;

NOR Ra, Rs	11	Rs	Rd	0010 0000	// NOR bitwise
					gr[Rd] <= ~ gr[Rd] | gr[Rs]
					z <= ~ |(~(gr[Rd] | gr[Rs]));
					pc <= pc+1;

NOP	11	111	111	1111 1110	// No OPeration pc <= pc+1;

HLT	11	111	111	1111 1111	// HaLT
					run <= 0;
					pc <= pc+1;

貳、Richard; Taylor; Zar

W.D. Richard、D.E. Taylor、D.M. Zar 在 1999 年發表的「A Capstone Computer Engineering Design Course.」[24]。Richard 等在所任教的美國華盛頓大學(Washington University)電子工程系(Electrical Engineering)發展 Computer Engineering 設計為專題的進階課程，課程內涵是以 VHDL 語言在 FPGA 上發展一個 8 Bits 的 microprocessor。此 microprocessor 的架構基礎為 Ivan Tomek 的著作"The Foundations of Computer Architecture and Organization"[12]，並作架構的擴展。實作的指令集如表 2-2 所示。

表 2-2 Richard 等 microprocessor 實作指令表

Group	Address Mode	Instructions											
6	Implied	ASR	LSR	SL	NEG	NOT	ROTL	ROTR	NOP				
1	Immediate	ADC	ADD	SUB	SBC	AND	OR	XOR	CMP	LD			
2	Absolute	ADC	ADD	SUB	SBC	AND	OR	XOR	CMP	LD	ST	LDDC	
0	Indirect	ADC	ADD	SUB	SBC	AND	OR	XOR	CMP	LD	ST		
3	Indexed	ADC	ADD	SUB	SBC	AND	OR	XOR	CMP	LD	ST		
7	Stack	ADC	ADD	SUB	SBC	AND	OR	XOR	CMP	LD/ POPAC	ST/ PUSHAC	POPF	PUSHF
5	I/O	IN	OUT										
4	Jump	JMP	JGE	JEQ/ JZ	JH	JC	JN	JV	JSUB	RET			

此課程目的在使學生經由現代化的設計流程、文字化的設計描述，完成 microprocessor 的設計專題。但學生必須具備 VHDL simulation、合成與繞線分析、路徑分析等先備知識，或已經習過計算機工程、邏輯設計、計算機結構、程式及軟體設計等相關課程。

參、Brown; Lomax; Carichner; Drake

R.B. Brown、R.J. Lomax、G. Carichner、A.J. Drake 在 2000 年所發表的「A Microprocessor Design Project in an Introductory VLSI Course.」[21]。其中提及密西根大學(University of Michigan)由 1980 年開始開設 VLSI Design 簡介的課程，在 1990 年課程內涵為 8 Bits Microprocessor 設計，於

1996 年再重新設計為 16 Bits RISC Microprocessor 設計。

Brown 等人認為 VLSI Design 課程的目的，在於培養學生具備建置一小型 Microprocessor 的能力，這將是學生未來成為 IC 設計師應具備的基礎能力之一，因此在課程內涵中強調學生透過使用各種 CAD 工具的設計經驗、測試方式，能立即得到強而有力的學習回饋，故而在 Microprocessor 的設計上，運作效能並非是最重要的考量依據，但在未來課程設計上，運作效能的改善亦能加入課程中做為進階設計探討的主軸。Brown 等人同時亦認為，計算機科技因設計工具的演進及 processor 架構的改變，課程內涵必須隨之因應與修正，以符合學生學習需求。

肆、Calazans; Moraes

N.L.V. Calazans、F.G Moraes 在 2001 年所發表的「Integrating the Teaching of Computer Organization and Architecture With Digital Hardware Design Early in Undergraduate Courses.」[20]。提出實施計算機組織與計算機結構課程，實作硬體設計的新方式。Calazans 與 Moraes 所任教的巴西 Rio Grande do Sul 大學(University of Rio Grande do Sul)，計算機科學系的開課內涵如圖 2-2 所示。

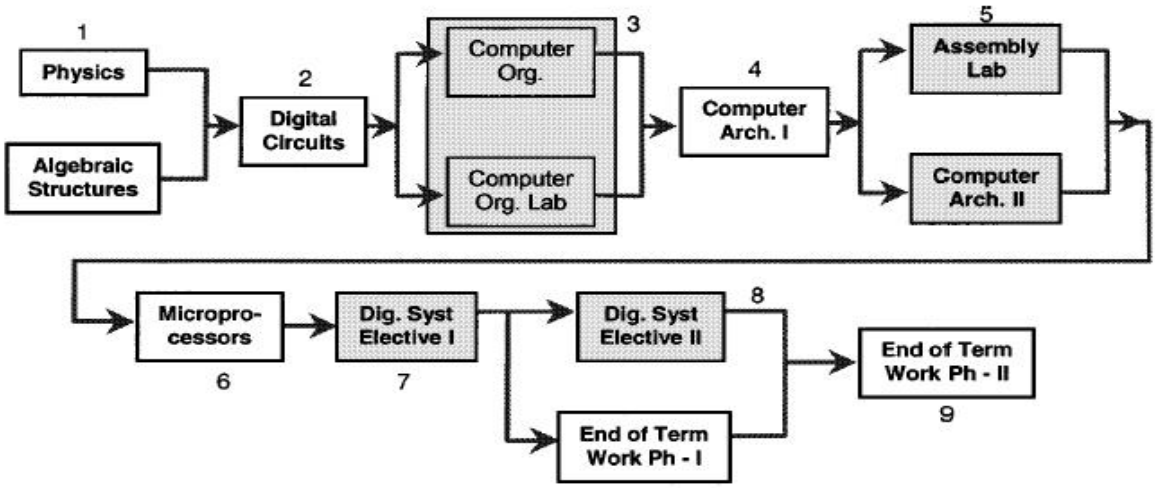

圖 2-2 巴西 Rio Grande do Sul 大學計算機科學系開課內涵

其中數字表示該課程開課的學期；箭頭表示課程與課程之間的流向，即欲修習該課程前所需預先修習的相關課程。在課程實施上，安排每週 4 小時的計算機組織教學，以及每週 2 小時的實驗課程。

計算機組織(Computer Org.)課程部分，著重於電腦架構原理探討，以及簡單的 Von Neumann 架構和 RTL 層次的抽象概念建立。計算機結構(Computer Arch.Ⅰ、Ⅱ)則將焦點置於使用 HDLs 語言進行更高層次的抽象設計、計算機結構效能分析、管線化(Pipelined)、I/O 系統、記憶體階層、記憶體管理(Memory Management Unit, MMU)、並行運算規則等方面的探討與實作。

Calazans 與 Moraes 在 CPU 設計方面使用繪製電路圖方式讓學生學習，採用 Xilinx Foundation 3.1i 為工具。最後再安排 VHDL 程式語言給學生做簡單的排序演算法、Test Bench 撰寫練習。最後，Calazans 與 Moraes 更設計了一份具 16 個問題的學習問卷做為教學成效驗證。應用 CPU 設計、FPGA、VHDL 在課程中，平均上而言學生學習的成效是相當顯著的。

伍、Pearson, Armstrong; McGregor

M. Pearson、D. Armstrong、T. McGregor 在 2002 年所發表的「Design of a processor to support the teaching of computer systems.」[19]。計算機系統教學包含了計算機結構、組合語言以及作業系統建置。在紐西蘭 Waikato 大學的計算機科學系(Department of Computer Science)使用 FPGA 自行發展 Computer System 教學的 CPU 及實驗板，並應用於教學上的經驗及過程，此 CPU 在指令集應用方面著重淺顯易懂，讓學生了解 CPU 架構及運作流程，而非考量 CPU 的運算效能。

在指令集結構方面，採用類似 MIPS CPU 的指令集結構，如圖 2-3 所示，其優點是 MIPS CPU 發展已久，已被認同為標準指令集架構，不需再費心思探討指令集架構的設計。另外在 CPU 設計模組區塊，如圖 2-4 所示，採用 VHDL 為設計基礎，並在自行設計的 FPGA 實驗板中下載驗證。

Waikato 大學的計算機科學系更逐步發展其他輔助工具，如 Compiler 及 Monitor 程式，形成完整的計算機結構教學課程。

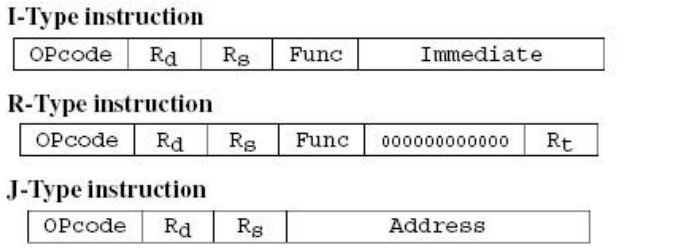

圖 2-3 Pearson 等 CPU 實作指令集格式

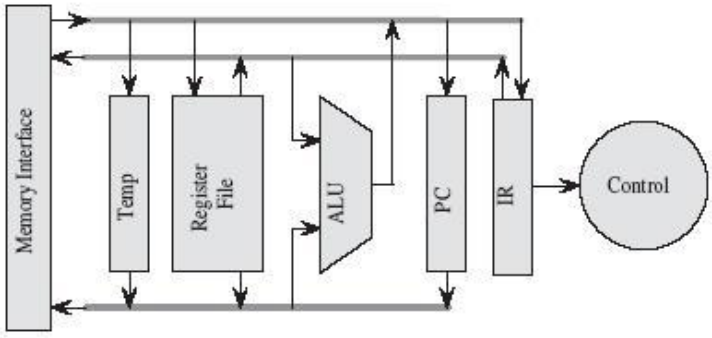

圖 2-4 Pearson 等 CPU 實作模組區塊

陸、總結

總合以上數篇計算機組織與設計的教學文獻，可由下列幾個構面，歸納出相關結論。

一、課程範圍

1. 本研究的課程範圍為核心課程「計算機組織與設計」，對照圖 2-2 為計算機組織後期之計算機結構設計與實作(Computer Arch. Ⅰ、Ⅱ)，並不包含組合語言(Assembly Lab)設計、高階語言轉換設計等。

2. 計算機組織與設計以往難於安排實作課程。計算機組織與設計實作電路規模大，使用傳統麵包板插接方式不僅難於完成且更難以偵錯、擴充，因此上述文獻皆使用 FPGA/CPLD 晶片來作計算機組織與設計實作課程，具有設計方式簡單、擴充性佳、可重覆規劃等等

的特性。

二、課程時數

1. 配合教學時數限制,規劃適當的課程內涵並包含理論教學與實作課程。各校對計算機組織與設計課程教學時數安排不一，以 Calazans、Moraes 文章中提到，安排 4 小時的理論課程及 2 小時的實作課程[20]。因此在有限的時間下，決定課程的廣度與深度並安排適用於各校之課程內涵絕對必要。

三、課程內涵

1. 在上述文獻中大多使用設計 8 Bits 或 16 Bits CPU 為主軸,而 32 Bits CPU 架構比 8 Bits 或 16 Bits 複雜許多,且 32 Bits CPU 是目前市場上的主流，因此導入 32 Bits CPU 設計為計算機組織與設計之實作課程較能符合實際需求。
2. 文獻中所實作的指令集少，應用範圍較小。
3. 由以上的文獻觀之，計算機結構課程的核心在於 CPU 架構的理論與實作。在實作方面，Ochi、Richard、Pearson 等人在其文章中皆有提到自製實驗平台以做驗證,故而發展自製的實驗平台應該整體課程極重要的一環。

此外，在 CPU 設計的部分，採用 VHDL、Verilog 語言是設計潮流之一。在 CPU 架構方面，為配合一週三 3 小時的教學時數限制下，由於 Pipelined RISC CPU 較難於理解、設計、偵錯並執行，因此本研究未採用 Pipelined 設計。而 MIPS CPU 發展已久，在架構與指令集方面發展亦趨成熟，若將課程內涵著重於設計與實作方面，不僅能配合課程時數的安排，再者由理論理解到實作過程中，更能將學生的設計經驗具體化。

在考量課程範圍、課程時數及課程內涵三項要素後，確立本研究之硬體平台需求，並著手規劃設計。