

AIoT 異質性嵌入式系統開發實戰系列

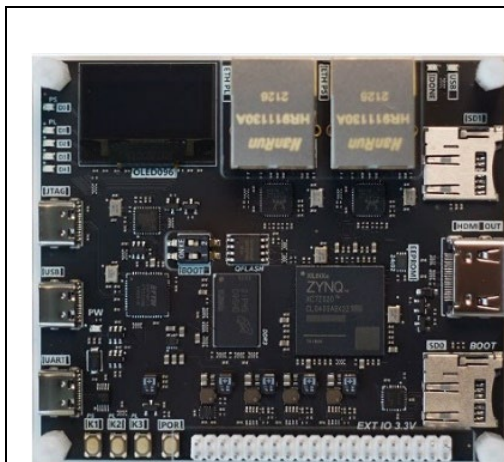
【嵌入式 FPGA 系統加速應用程式設計】

使用 Xilinx ZYNQ 7020 (支持 PYNQ)開發板教學

工研院產業學院特邀擁有多年實務開發經驗之專業講師進行授課，課程內容強調以業界實務為導向，著重於程式設計實務之演練，除了以教學投影片講解程式設計理論及語法，同時將於課堂上現場撰寫程式範例，並直接於教學板上執行測試，讓學員清楚了解如何從無到有，完整的撰寫程式，望能培養學員全方位完整系統開發與符合業界需求之能力。

單元	課程日期	課程名稱	時數
—	3/27、4/3、 4/10、4/17(三)	FPGA Verilog HDL 數位邏輯電路設計與周邊控制 實戰-使用 Xilinx Vitis/Vivado	28
— —	5/8、5/15、 5/22、5/29(三)	FPGA 設計與圖像處理實戰 -使用 Vitis/Vivado HLS	28

❖ 教學使用開發版：



▲ 開發版示意圖 ▲

主晶片型號：XC7Z020-CLG400-2
 板載晶振：50M(PL)、33.33M(PS) 雙晶振
 板載 1000M 乙太網 RJ45 接口:PS*1、PL*1 雙網口
 板載 Xilinx JTAG 下載調試器,支持 FPGA 及 ARM 全部開發
 板載串口 TYPE C 接口(PS)
 板載雙 SD 卡,SD0 及 SD1,SD0 支持啟動
 板載 HDMI(PL)接口
 板載 0.96" OLED(PL)
 板載 W25Q128FVSG 128Mbit FLASH
 板載 PL 按鍵*2、PS 按鍵*1,PL LED*4、PS LED*1
 擴展 IO 34 個接口,支持各種實驗

❖ 培訓證書：總出席率達 80%，將由工業技術研究院產業學院核發培訓證書。

本課程規劃「線上同步數位學習」形式，同步線上學習不受地點限制、增進專業能力！

單元一：FPGA Verilog HDL 數位邏輯電路設計與周邊控制實戰

在 FPGA 數位電路設計上超大型積體電路設計(VLSI)或系統晶片(SoC)設計均扮演著非常重要的角色，是軟硬體工程師在職場上必備的工具。工業 4.0(生產力 4.0)應用以機器手臂為重要抓手的“工業 4.0”戰略，會加速智慧製造的進程。未來物聯網與工業 4.0 是大勢所趨，而高性能電機控制是實現工業 4.0 的關鍵應用。使用 FPGA/Verilog 可充分滿足當今磁場定向控制(FOC)等複雜控制演算法所提出的苛刻的時序和性能要求。

本課程搭配 FPGA 實驗板進行，先詳細講述 FPGA / Verilog HDL 語法與應用，採深入淺出講解並輔以實作 DEMO 方式進行實戰教學，並結合【IP-Core 設計應用】以加速產品開發速度。

將完整解說 Verilog HDL 語法及其對應之數位邏輯電路之呈現，課程後能熟悉 Verilog 語言全貌，帶領大家進入以 Verilog 為主的各種相關設計領域，因此非常適合各層次的設計者參考使用，上完課程後可直接與業界銜接，真正提升數位邏輯電路設計能力。

❖ 課程特色：

1. 目前業界 FPGA / CPLD 的主流廠家為 AMD Xilinx，相關 TOOL 操作廠商與代理商都會提供良好的訓練。所以本課程的進行，是設計語法與實作並重，透過範例與各種介面專題之設計，讓學員在以 Verilog 為主的各種相關設計領域中，熟悉 Verilog 語言全貌，大大的提升學員的數位設計與 IC 設計能力。
2. 課程內容進行，將由講師實際講解 Verilog Coding 的實用密技技巧之心法，讓學員心中有數位邏輯電路，手中可寫出對應 Verilog 程式代碼。
3. 帶領學員以數位邏輯電路之思維方式，熟知 Verilog 程式不再是一行行的代碼，而是一塊一塊的硬體模組，進而達到在 FPGA 處理速度與面積間，獲得最佳效果。

❖ 適合對象：

1. 想從事 CPLD、FPGA 相關研發工作者。
2. 資訊、資工、資管、電子、電機等相關科系畢業生。
3. 熟悉邏輯設計以計算機組織者佳。

★【基礎 LAB】

Verilog 循序電路設計：正反器、暫存器(Flip-Flop)、LED 或七段顯示器(一位數與多位數七段顯示器)、按鍵控制電路、計時器(Timer)/毫秒、計數器(Counter)電路實驗、Finite State Machine：比大小遊戲電路實驗、紅綠燈控制電路實驗

★【進階 LAB】

Simple 4 bit ALU implementation、PLL 電路實驗、LCD1602 實驗、RS-232 通訊(UART)控制電路實驗、I2C/SPI 系統電路實驗(24C08 讀寫 LAB)、PWM 控制電路實驗(Servo Controller)、VGA 控制電路實驗

序	日期	課程內容
1	113/03/27 (三) 9:30-17:30	(1) FPGA 工具介紹、安裝與使用及如何使用電路模擬：介紹 AMD Xilinx 之 FPGA/CPLD 開發工具 Vitis / Vivado (2) 階層模組觀念(Hierarchical Modeling Concepts)設計方法 (3) Verilog HDL 語法協定(Lexical conventions)：資料型態(data types)、記憶體(Memories)、系統任務(system tasks)、編譯命令(compiler directives) (4) Verilog HDL 模組與輸出入埠(Modules and Ports)：模組(Module definition)、輸出入埠(port declaration, connecting ports)、階層化取名(hierarchical name referencing) (5) 邏輯合成(Logic Synthesis)、邏輯閘層次模型(Gate-Level Modeling)
2	113/04/03 (三) 9:30-17:30	(1) Verilog HDL 資料處理模型(Dataflow Modeling)：持續指定(Continuous assignments)、延遲(delay specification)、運算式(expressions)、運算子(operators)、運算元(operands)、運算子種類(operator types) (2) Verilog HDL 行為模型(Behavioral Modeling)：結構化程序(Structured procedures)、initial 及 always 敘述、程序指定(阻礙指定 blocking 及無阻礙指定 nonblocking statements)、時序控制(delay control, event control)、條件敘述(conditional statements)、多路徑分支(multiway branching)、迴圈(loops)、循序與平行區塊(sequential and parallel blocks) (3) Verilog HDL 任務(Tasks)與函數(Functions)：函數(Functions)、任務(Tasks)、Assignment
3	113/04/10 (三) 9:30-17:30	(1) 演算法狀態機(Algorithmic State Machine)、有限狀態機(Finite State Machine)：Moore FSM、Mealy FSM and Modeling。 (2) Verilog HDL 時序及遲延(Timing and Delays)、Verilog HDL 時序仿真(Timing Simulation) (3) Verilog generate / endgenerate statement (genvar、localparam、generate if、generate for、generate case) (4) Verilog HDL 有用之程式技巧(Useful Modeling Techniques)：程序持續指定(assign 與 deassign 及 force 與 release)、參數(parameter) and 複寫參數(defparam, module instance)、條件式的編譯與執行、時間刻度(Time Scales)、有用的系統任務: File (\$fopen,\$fdisplay,...)、隨機亂數產生器(Random Number Generation)、VCD File
4	113/04/17 (三) 9:30-17:30	(1) FPGA 晶片內建記憶體 (In-System Memory)、Memory Synchronous/Asynchronous Read/Write、Synchronous FIFO and Asynchronous FIFO (2) 【 IP-Core 設計應用 】：Verilog IP-Core(Intellectual Property Core)<<for AMD Xilinx>> <ul style="list-style-type: none"> ● 添加設計和發布 IP-Core (Adding design and publish IP-Core) ● 如何添加 AMD Xilinx 的 IP-Core (How to adding AMD Xilinx's IP-Core: such as General-purpose input/output, PLL, Timer, RAM, ROM, UART, I2C, SPI...) ● 如何移植 OpenCores (How to Porting OpenCores: such as UART IP) (3) 設計除錯(Design Debugging)：ILA IP、Timing Analyzer

※ 因不可預測之突發因素，主辦單位得保留課程之變更權利。

單元二：FPGA 設計與圖像處理實戰

帶領學員了解各種 AMD Xilinx Vision / xfOpenCV 計算機視覺問題原型，及深入了解計算機視覺 Vitis vision Library，能獨立正確處理更深入的計算機視覺應用，如人臉識別、物體檢測等。

❖ 課程特色：

1. 使用 Vitis / VIVADO HLS 和 FPGA 進行圖像處理：利用 HLS 上的計算機視覺和圖像 / 視頻處理庫。
2. 在帶有 HLS IP 的 Vitis / VIVADO 工具上設計完整的圖像處理流水線，並在 Zynq FPGA 測試設計。
3. 創建 C/C++ Project，使用高級合成(Vitis/VIVADO HLS)對其進行模擬、合成和導出。
4. 調試和優化 AMD Xilinx FPGA HLS Project。

❖ 適合對象：

1. 想從事 CPLD、FPGA 相關研發工作者。
2. 資訊、資工、資管、電子、電機等相關科系畢業生。
3. 熟悉邏輯設計及計算機組織者佳。

序	日期	課程內容
1	113/05/08 (三) 9:30-17:30	(1)AMD Xilinx Vision / xfOpenCV 簡介 (2)Using the Vitis vision Library (3)Getting Started with Vitis Vision(xfOpenCV): Vitis Design Methodology & Host Code with OpenCL (4)Introduction Wrappers around HLS Kernel(s) & xf::cv::Mat class (5)Introduction Stream Based Kernels: xf::cv::Array2xfMat()、xf::cv::xfMat2Array() (6)Vitis vision Library API Reference、圖像的基本操作(Basic Operations on Images) (7)First Vitis Vision Application: Dilation Accel、Design Examples Using Vitis Vision Library
2	113/05/15 (三) 9:30-17:30	(1)重要 GUI 操作：圖像的算術運算(Arithmetic Operations on Images)、效能測量和改進技術(Performance Measurement and Improvement Techniques) (2)圖像處理(Image Processing)-1：更改顏色空間(Changing Colorspaces)、圖像的幾何變換(Geometric Transformations of Images)、圖像閾值(Image Thresholding)、
3	113/05/22 (三) 9:30-17:30	(1)圖像處理(Image Processing)-2：平滑圖像(Smoothing Images)、形態轉換(Morphological Transformations)、圖像漸變(Image Gradients)、圖像金字塔(Image Pyramids)、輪廓(Contours)、直方圖(Histograms)、圖像變換(Image Transforms)、模板匹配(Template Matching) (2)特徵檢測和描述(Feature Detection and Description)
4	113/05/29 (三) 9:30-17:30	(1)視頻分析(Video Analysis) (2)機器學習(Machine Learning)：支持向量機(Support Vector Machines (SVM)) (3)計算攝影(Computational Photography)：圖像去噪(Image Denoising)、圖像修復(Image Inpainting) (4)物體檢測(Object Detection) (5)L2 Canny 邊緣偵測介紹

※ 因不可預測之突發因素，主辦單位得保留課程之變更權利。

❖ 講師簡介：江老師

學 歷：中正大學資訊工程所博士班研究、逢甲大學資訊工程研究所畢

專 長：嵌入式系統開發、智慧型嵌入式家庭自動化系統設計、8051 單晶片、多套自
動化系統及驅動程式 Driver、WEB-ERP 系統、ERP 系統、生產線自動化系統。

作 品：1. MX6/8 產品開發設計、S3C6410 開發板研發設計、STM32F7/H7 產品設計
2. 智慧型嵌入式家庭自動化系統
3. 8051 單晶片於工業機台之自動化系統
4. 多套自動化系統(群錄自動化)及驅動程式 Driver
5. WEB-ERP 系統、多套 ERP 系統
6. 華映公司生產線自動化系統
7. 手機遊戲程式(象棋、打磚塊)
8. 國內第一套網路中文傳呼系統

【課程辦理資訊】

★自備物品-筆記型電腦★建議使用 Windows 7 以上、Mac 2011 或更高階的作業軟體。筆記型電腦規格 CPU I5 或同等級以上、硬碟需求容量 256G(含)以上、RAM 8G(含)以上、具 Wifi 或網路連網功能。

❖ 上課地點：工研院產業學院 台北學習中心。 實際地點依上課通知為準!

❖ 課程聯絡人：(02)2370-1111 分機 310 · itri535166@itri.org.tw 陳小姐。

❖ 課程費用：**數位與實體價格一致**

方案	一般報名	早鳥優惠 (課前三週報名且繳費)	三人以上團報優惠
每單元(28 hrs)	25,500/人	23,000/人	21,700/人
全系列優惠(56hr)	原價 51,000 元 · 全系列優惠 43,400/人		

❖ 系列課程推薦：

課程名稱	單元名稱	時數	課程日期
嵌入式 ARM Cortex-M7 及 RTOS 應用開發	嵌入式物聯網 ARM Cortex-Mx 系統韌體開發	28	1/19、1/24、1/26、 1/31(三、五)
	嵌入式 RTEMS RTOS 硬即時作 業系統移植與開發	21	3/8、3/13、3/15(三、五)
嵌入式物聯網 Linux 系統開發 工程師培訓班 (假日班)	嵌入式 Linux 系統實作與程式 設計實務	21	3/9、3/16、3/23(六)
	嵌入式 Linux 驅動程式實務	21	4/13、4/20、4/27(六)
	嵌入式 IoT Linux 網路通訊及 多媒體應用實務	21	5/11、5/18、5/25(六)
嵌入式 FPGA 系統加速應用 程式設計	FPGA Verilog HDL 數位邏輯電 路設計與周邊控制實戰-使用 Xilinx Vitis/Vivado	28	3/27、4/3、4/10、4/17 (隔週三)
	FPGA 設計與圖像處理實戰 -使用 Vitis/Vivado HLS	28	5/8、5/15、5/22、5/29 (隔週三)

報名表

FAXTO : (02)2381-1000 陳小姐收

【嵌入式 FPGA 系統加速應用程式設計】

勾選	模組	課程名稱	時數	開課日期
<input type="checkbox"/> 實體 <input type="checkbox"/> 數位	—	FPGA Verilog HDL 數位邏輯電路設計與 周邊控制實戰-使用 Xilinx Vitis/Vivado	28	3/27、4/3、4/10、 4/17(隔週三)
<input type="checkbox"/> 實體 <input type="checkbox"/> 數位	— —	FPGA 設計與圖像處理實戰 -使用 Vitis/Vivado HLS	28	5/8、5/15、5/22、 5/29(隔週三)
公司全銜		統一編號		
發票地址		傳 真		
參加者姓名	部 門	電 話	手 機	E-mail
		()		
		()		
聯絡人		()		

- 信用卡 (線上報名) : 繳費方式選「信用卡」, 直到顯示「您已完成報名手續」為止, 才確實完成繳費。
- ATM 轉帳 (線上報名) : 繳費方式選擇「ATM 轉帳」者, 系統將給您一組轉帳帳號「銀行代號、轉帳帳號」, 但此帳號只提供本課程轉帳使用, 各別學員轉帳請使用不同轉帳帳號!! 轉帳後, 寫上您的「公司全銜、課程名稱、姓名、聯絡電話」與「收據」回傳。
- 銀行匯款(公司逕行電匯付款) : 土地銀行 工研院分行, 帳號 156-005-00002-5(土銀代碼: 005)。戶名「財團法人工業技術研究院」, 請填具「報名表」與「收據」回傳。
- 即期支票: 抬頭「財團法人工業技術研究院」, 郵寄至: 106台北市大安區復興南路二段237號4樓, 陳小姐收。
- 計畫代號扣款(工研院同仁) : 工研院員工報名請網路點選「工研人報名」填寫計畫代號後, 經主管簽核同意即可。

- 1、請註明服務機關之完整抬頭, 以利開立收據; 未註明者, 一律開立個人抬頭, 恕不接受更換發票之要求。
- 2、若報名者不克參加者, 可指派其他人參加, 並於開課前一日通知。
- 3、如需取消報名, 請於開課前三日以書面傳真至主辦單位並電話確認申請退費事宜。逾期將郵寄講義, 恕不退費。