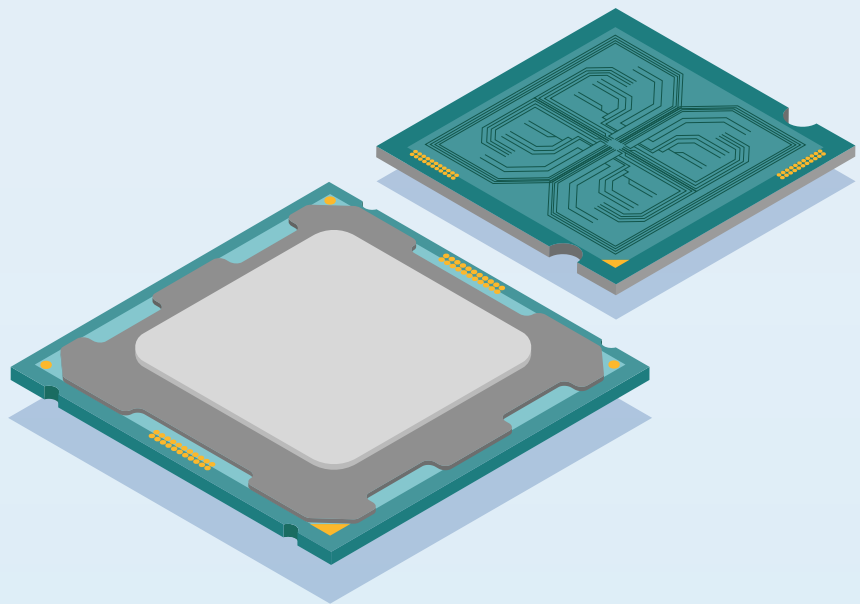


เรียนรู้การออกแบบ FPGA สำหรับ IoT



โดย ชำนาญ ปัญญาใส

ทีมระบบไซเบอร์-กายภาพ (CPS)
หน่วยทรัพยากรด้านการคำนวณและไซเบอร์-กายภาพ (NCCPI)
ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (เนคเทค)

เรียนรู้การออกแบบแอปพลิเคชันสำหรับ IoT

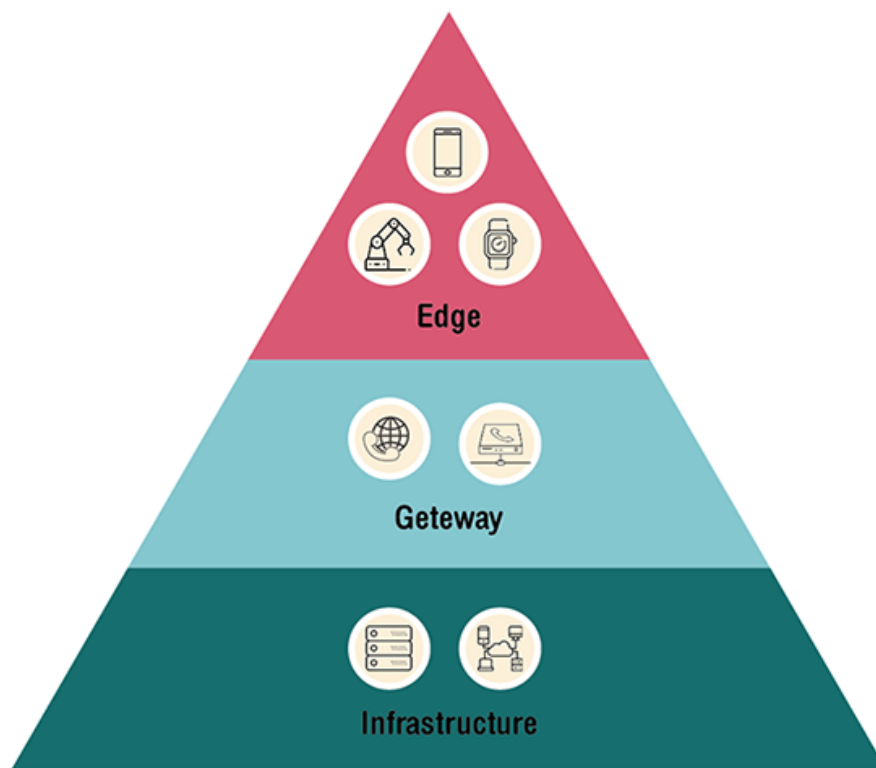
โดย ชำนาญ ปัญญาใส อีเมล: chumnarn@nectec.or.th

บทนำ

ในช่วงหลายปีที่ผ่านมา กระแสเทคโนโลยีอินเทอร์เน็ตของสรรพสิ่ง หรือ IoT (Internet of Things) เข้ามามีผลกระทบต่อการดำเนินชีวิตของคนเราจนถึงการดำเนินการธุรกิจต่างๆ ที่เกิดจากการที่เราสามารถเชื่อมต่ออุปกรณ์ต่างๆ ผ่านเน็ตเวิร์คได้ทั่วโลก มีการส่งผ่านข้อมูลจากอุปกรณ์หน่วยย่อยๆ เช่น เซนเซอร์ขึ้นไปเก็บที่ศูนย์ข้อมูลในรูปแบบคลาวด์ เพื่อใช้วิเคราะห์ประมวลผลต่างๆ

สถาปัตยกรรมของระบบ IoT จึงถูกแบ่งเป็น 3 ส่วนหลักๆ ประกอบด้วย

- Edge เป็นส่วนอุปกรณ์ปลายทางที่เชื่อมต่อโลกภายนอกเข้าสู่ระบบเครือข่าย เช่น อุปกรณ์เซนเซอร์ กล้องถ่ายภาพ อุปกรณ์ actuator (เช่น อุปกรณ์สวิตช์ อุปกรณ์รีเลย์ อุปกรณ์ PLC) การส่งผ่านข้อมูลในรูปแบบโปรโตคอลสื่อสาร เช่น I/O Link หรือ ModBus หรือ Ethernet หรือ SmartBUS หรือ CAN (EU)
- Gateway เป็นอุปกรณ์ทำหน้าที่เชื่อมต่อระหว่างอุปกรณ์ปลายทาง รวมข้อมูลจากอุปกรณ์ปลายทาง ส่งผ่านข้อมูลเข้าสู่อินเทอร์เน็ตไปยังชั้นลำดับถัดไป นอกจากนี้ยังทำหน้าที่ในการแปลงโปรโตคอลข้อมูลสื่อสาร ทำการประมวลผลและกรองข้อมูล
- Infrastructure ชั้นโครงสร้างพื้นฐาน ซึ่งรวมถึงบริการต่างๆ ในรูปแบบคลาวด์ การประมวลผลข้อมูล และแพลตฟอร์มแอปพลิเคชันต่างๆ รูปแบบบริการคลาวด์ เช่น Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS) ตัวอย่าง ผู้ให้บริการคลาวด์ ด้าน IoT เช่น Artik Cloud, Autodesk Fusion Connect, AWS IOT, GE Predix, Google Cloud IoT, Microsoft Azure IoT Suite, IBM Watson IoT, ThingWorx, Intel IoT Platform, Salesforce IoT Cloud, Telit DeviceWise, Zebra Zatar Cloud, macchina.io, ThingSpeak, Particle Cloud, NETPIE



รูปที่ 1 แสดงโครงสร้างสถาปัตยกรรมของระบบ IoT

เอฟพีจีเอกับ IoT

จากที่กล่าวมาข้างต้น จะเห็นภาพองค์ประกอบของเทคโนโลยี IoT ซึ่ง บทความนี้จะกล่าวถึงเทคโนโลยีเอฟพีจีเอ (FPGA: Field Programmable Gate Array) เพื่อการประยุกต์ใช้งานในส่วนต่างๆ ของระบบ IoT โดยในเบื้องต้นจะได้อธิบายพื้นฐานของเทคโนโลยี และพื้นฐานการออกแบบเอฟพีจีเอสำหรับผู้เริ่มต้นพัฒนาและประยุกต์ใช้งาน

ในการพัฒนาอุปกรณ์และแก้ปัญหาโจทย์ด้าน IoT สามารถมีได้หลายแนวทาง เช่น ใช้อุปกรณ์ ไมโครคอนโทรลเลอร์ (MCU) ไมโครโปรเซสเซอร์ (CPU) หน่วยประมวลผลกราฟิก (GPU) หน่วยประมวลผลสัญญาณดิจิทัล (DSP) และอุปกรณ์เอฟพีจีเอเป็นอีกทางเลือกที่เข้ามามีบทบาทอย่างมาก ด้วยคุณสมบัติเด่นของเอฟพีจีเอ เช่น

- การทำงานประมวลผลแบบขนาน ที่ทำให้สามารถอ่านข้อมูลจากเซนเซอร์จำนวนมากได้พร้อมกัน
- เอฟพีจีเอสามารถที่จะโปรแกรมวงจรฮาร์ดแวร์ภายในใหม่ได้ ดังนั้นจึงไม่จำเป็นต้องปรับแก้ไขฮาร์ดแวร์ภายนอก เมื่อมีการปรับแก้ไขวงจร
- การกินไฟค่อนข้างต่ำ ทำให้เกิดการใช้พลังงานอย่างมีประสิทธิภาพ

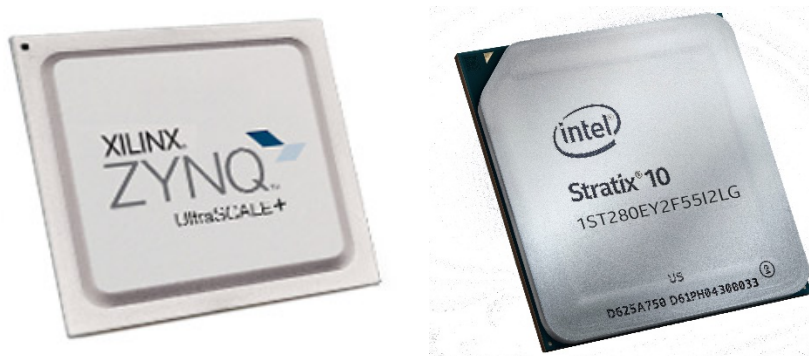
ด้วยธรรมชาติของเอฟพีจีเอที่เราสามารถทำการโปรแกรมใหม่ได้ ทำให้มีความยืดหยุ่นในการประยุกต์ใช้งาน ได้หลากหลาย เช่น

- Aerospace & Defense
- ASIC Prototyping
- Audio
- Automotive
- Broadcast & Pro AV
- Consumer Electronics
- Data Center
- High Performance Computing and Data Storage
- Industrial
- Medical
- Security
- Video & Image Processing
- Wired Communications
- Wireless Communications

ตัวอย่างการใช้งานเอฟพีจีเอ ในส่วนของ Edge เช่นการใช้งานเอฟพีจีเอในการประมวลผลอัลกอริทึมที่ซับซ้อน สำหรับอุปกรณ์อิเล็กทรอนิกส์ติดตัวในการตรวจจับการทกล้ม [1] หรือการใช้เอฟพีจีเอในการประมวลผลทางด้าน AI บนเซิร์ฟเวอร์ Azure ของไมโครซอฟต์ [2]

เทคโนโลยีเอฟพีจีเอ

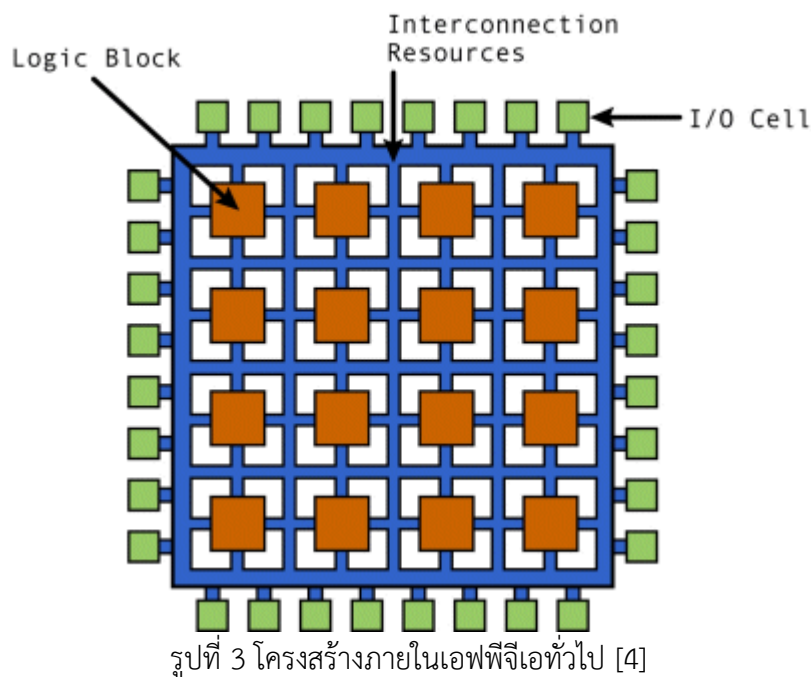
เอฟพีจีเอเป็นอุปกรณ์ไอซีที่ผู้ใช้สามารถทำการโปรแกรมวงจรภายในได้เอง ซึ่งผู้ใช้งานหรือผู้ออกแบบจะทำการออกแบบโดยการเขียนโค้ดภาษาสำหรับออกแบบฮาร์ดแวร์ขั้นสูง (High level hardware description language) เช่นภาษา VHDL [3] หรือ Verilog หรือ SystemVerilog ปัจจุบันนี้ ยังสามารถออกแบบฮาร์ดแวร์โดยใช้ภาษา C ได้อีกด้วย



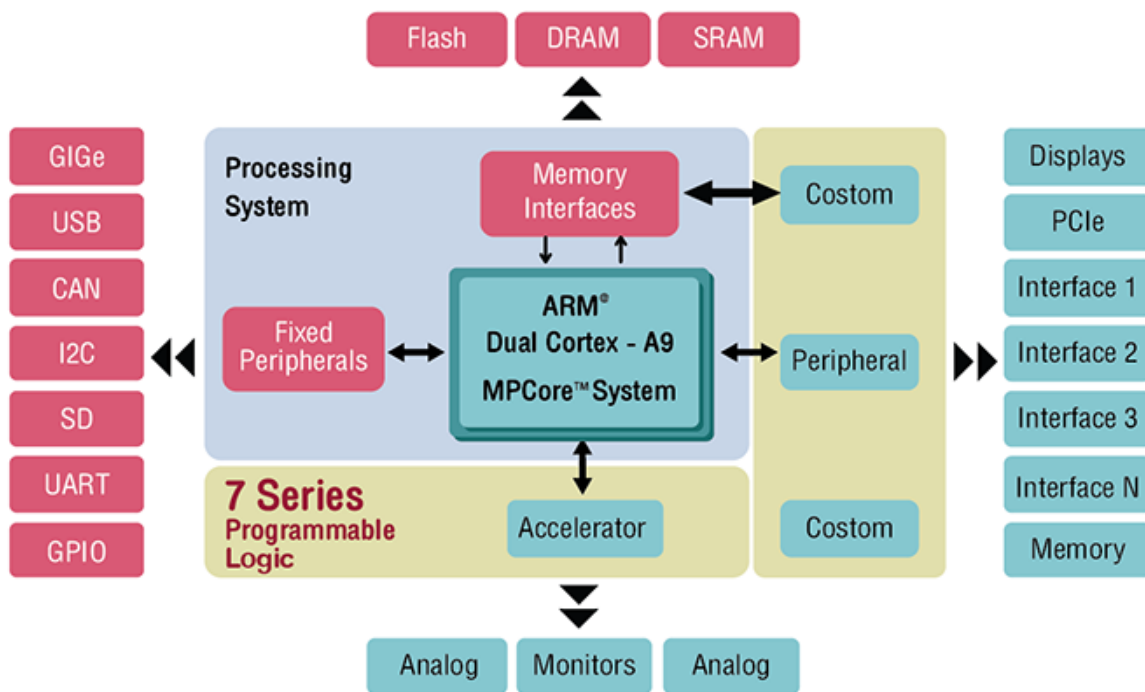
รูปที่ 2 ตัวอย่าง ไอซีเอฟพีจีเอ ของบริษัท Xilinx และบริษัท Intel

โครงสร้างภายในไอซีเอฟพีจี ประกอบด้วย

- Logic Module เป็นส่วนที่ใช้สำหรับทำหน้าที่ฟังก์ชันทางลอจิกต่างๆ และอาจจะมีหน่วยเก็บความจำในรูปแบบ Flip-Flop (FF) หรือ BlockRAM
- Programmable Interconnect เป็นส่วนสำหรับใช้เชื่อมต่อสัญญาณระหว่าง Logic Module ต่างๆ เข้าด้วยกัน
- Programmable I/O เป็นส่วนของขาสำหรับการเชื่อมต่ออินพุต-เอาพุต กับโลกภายนอก



ปัจจุบัน ไอซีเอฟพีจี ได้มีการพัฒนามากยิ่งขึ้น ได้มีการรวมเอาไมโครโปรเซสเซอร์และส่วนวงจรเสริมต่างๆ ไว้ภายในตัวไอซีเดียวกัน ซึ่งทำให้ขยายการประยุกต์ใช้งานได้หลากหลายยิ่งขึ้น ตัวอย่าง เช่น ไอซีเอฟพีจี Xilinx Zynq-7000 หรือ Intel Arria V ได้รวมเอา dual-core ARM Cortex-A9 MPCore ไว้



รูปที่ 4 Xilinx Zynq-7000 All Programmable System on a Chip (Source: www.xilinx.com)

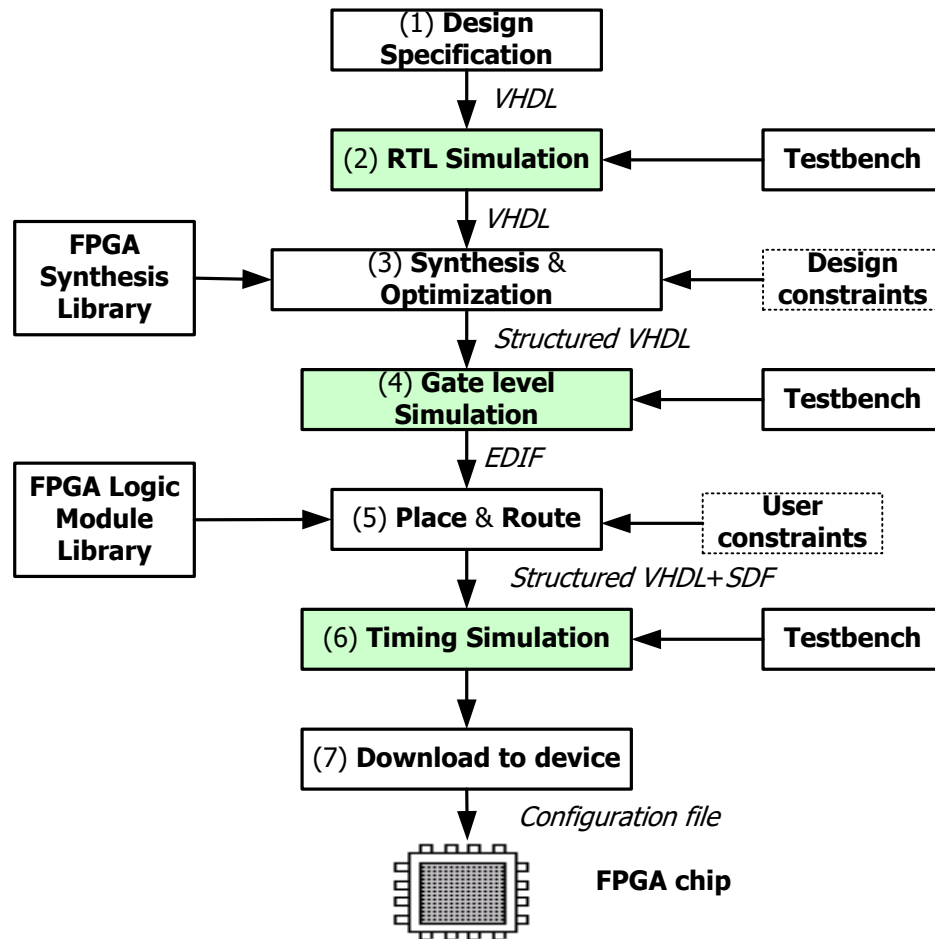
ผู้ผลิตและจำหน่ายอุปกรณ์เอฟพีจีเอ ในตลาด ได้แก่

- Xilinx (www.xilinx.com)
- Intel (www.intel.com)
- Microsemi (www.microsemi.com)
- Lattice Semiconductor (www.latticesemi.com)
- Quicklogic (www.quicklogic.com)
- Achronix Semiconductor (www.achronix.com)

โดยที่ Xilinx และ Intel มีส่วนแบ่งตลาดรวมประมาณ 90 เปอร์เซ็นต์

การออกแบบด้วยเอฟพีจีเอ

ขั้นตอนการออกแบบวงจรดิจิทัลด้วยเอฟพีจีเอทั่วไป สามารถแสดงได้ดังรูปที่ 5

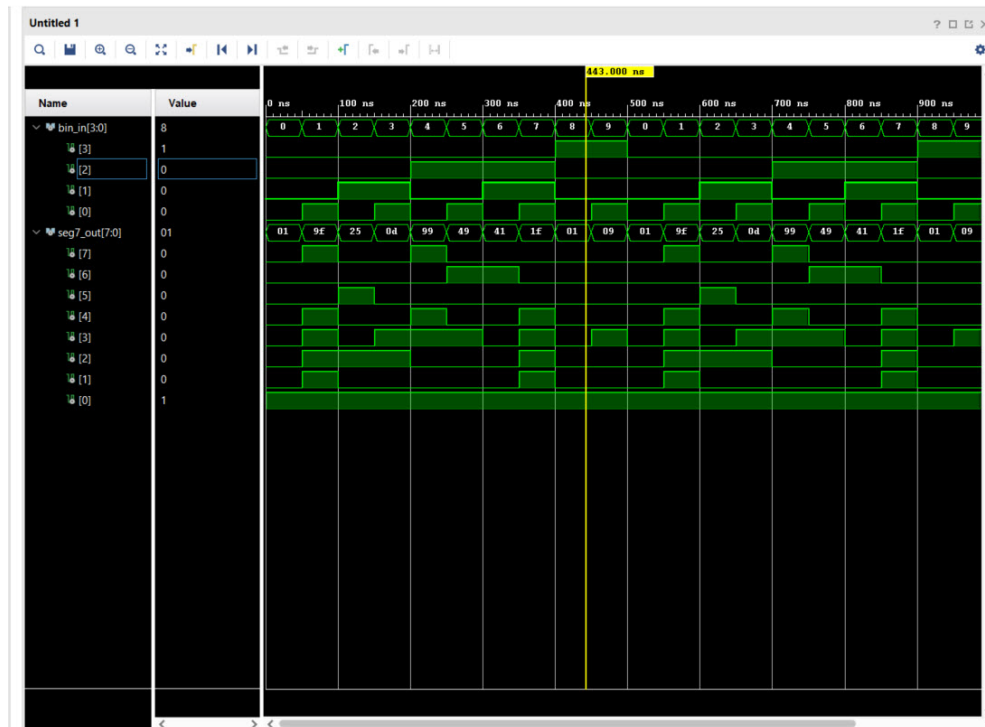


รูปที่ 5 แสดง FPGA Design Flow โดยใช้ภาษา VHDL

โดยจะมีขั้นตอนหลักๆ ประกอบด้วย

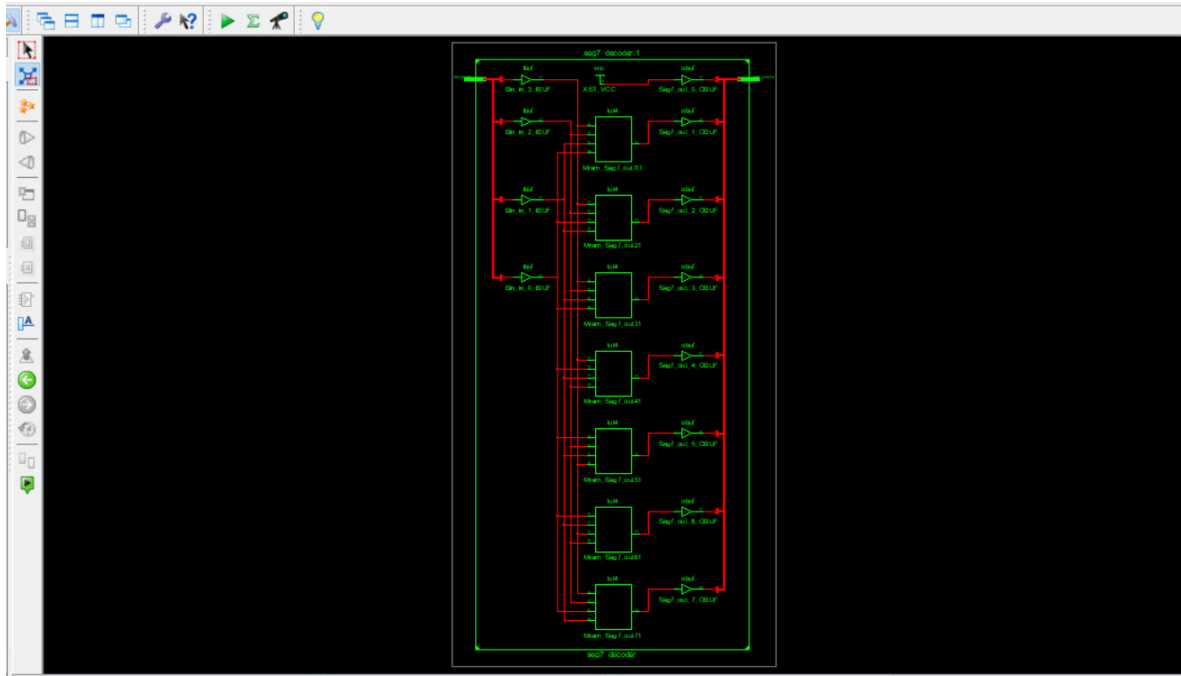
1. Design Specification เป็นการกำหนดรายละเอียดของวงจรที่ต้องการ ซึ่งจะบ่งบอกถึงส่วนประกอบต่างๆ ของวงจร ส่วนอินพุต เอาท์พุต ความเร็วในการทำงาน อัตราการกินไฟ ฯลฯ จากข้อกำหนดต่างๆ นี้จะสามารถนำมาเขียนเป็นผังวงจร (schematic) หรือ เขียนโค้ดภาษา VHDL หรือ Verilog หรือ SystemVerilog ได้อีกด้วย

2. RTL Simulation เป็นขั้นตอนการจำลองการทำงานของโค้ด VHDL หรือ Verilog ในระดับ RTL (Register Transfer Level) ในการจำลองการทำงาน อาจจะใช้การสร้างสัญญาณอินพุตที่เป็นกราฟ (waveform) หรือถ้าวงจรที่ซับซ้อนมากอาจจะมีการเขียน testbench เพื่อสร้างสัญญาณป้อนเข้าไปยังวงจรที่ต้องการทดสอบ



รูปที่ 6 ภาพแสดงผลการจำลองการทำงานวงจรด้วยโปรแกรม ISim ของ Xilinx

3. Synthesis & Optimization ขั้นตอนการสังเคราะห์วงจรฮาร์ดแวร์ที่เกิดจากโค้ด VHDL (หรือ Verilog) ขั้นตอนการสังเคราะห์วงจรประกอบด้วยการแปลงโค้ดให้เป็นวงจรลอจิก (translate) แล้วทำการเทียบวงจรเข้ากับเทคโนโลยี (technology mapping) และขั้นตอนการ optimization วงจรให้ได้ตามข้อกำหนด ซึ่งถูกตีกรอบไว้โดยส่วนที่เรียกว่า design constraint แล้วทำการ save ผลการสังเคราะห์กลับเป็นไฟล์ VHDL ที่เป็นโครงสร้าง (structured VHDL) สำหรับการจำลองระดับเกต (Gate Level Simulation) และไฟล์ netlist ในรูปแบบไฟล์ EDIF (Electronic Design Interchange Format) สำหรับใช้ในการทำ Place & Route ในการสังเคราะห์จะมีการเลือกเทคโนโลยีเป้าหมายที่ใช้สำหรับการสังเคราะห์วงจร หรือเอฟพีจีเอที่ต้องการ (FPGA synthesis library)



รูปที่ 7 แสดงผลวงจรที่ได้จากการสังเคราะห์จากโค้ดภาษา VHDL

4. Gate Level Simulation หรือ post synthesis simulation เป็นขั้นตอนการจำลอง VHDL ระดับเกต ที่เกิดจากการสังเคราะห์วงจร เพื่อดูการทำงานอยู่ในข้อกำหนดเริ่มต้นหรือไม่ ด้วยการนำค่าความหน่วงของอุปกรณ์หรือลอจิกเกต (gate delay) เข้ามาคิดด้วย โดยมีการคิดค่าความหน่วงของสายสัญญาณ (interconnect delay) เท่ากับหนึ่งหน่วย จะเห็นว่าการทำงานของวงจรจะใกล้เคียงกับฮาร์ดแวร์จริงมากขึ้น

5. Place & Routeหรือบางทีอาจจะเรียกว่าขั้นตอน implementation ซึ่งเป็นการนำเอาวงจรลอจิกในรูปแบบไฟล์ EDIF (Electronic Design Interchange Format) ที่ผ่านการ optimization แล้วมาจัดวางบนชิปเอฟพีจีเอ และเกิดการเชื่อมต่อสัญญาณระหว่างส่วนลอจิกต่างๆ เสมือนกับว่ามีการวางตำแหน่งของลอจิกจริงๆ จากนั้นซอฟต์แวร์จะทำการคำนวณค่าความหน่วงของส่วนลอจิก (logic delay) พร้อมทั้งค่าความหน่วงของสายสัญญาณ (interconnect delay) จริงออกมา ในรูปแบบของไฟล์ SDF (Standard Delay Format) ซึ่งจะนำไปใช้สำหรับการจำลองการทำงานทางเวลา (timing simulation)

6. Timing Simulation หรือ post place & route simulation เป็นการจำลองการทำงานของวงจรที่ได้จากการสังเคราะห์ (structured VHDL) โดยมีการคิดค่าความหน่วงของเกต (gate delay) และความหน่วงการเชื่อมต่อที่เกิดขึ้นจริง (interconnect delay) ในการกำหนดการวางลอจิกและการเชื่อมต่อสัญญาณในเอฟพีจีเอซึ่งอยู่ในรูปแบบไฟล์ SDF ผลลัพธ์ที่เกิดขึ้นจะใกล้เคียงกับฮาร์ดแวร์จริง

7. Download to device หรือขั้นตอน Programming ขั้นตอนการโปรแกรมไฟล์บิต (bit file) หรือ configuration file ลงบนเอฟพีจีเอ วิธีการโปรแกรมอาจทำได้ 2 วิธี ได้แก่การดาวน์โหลดไฟล์บิตจากคอมพิวเตอร์ลงบนบอร์ดเอฟพีจีเอโดยผ่านโปรแกรม iMPACT หรือใช้วิธีโปรแกรมไฟล์บิตลงบนหน่วยเก็บความจำแบบแฟลช (flash memory) หรือ EPROM หรือ EEPROM

ซอฟต์แวร์สำหรับการออกแบบ

ซอฟต์แวร์ที่ใช้สำหรับการออกแบบเอฟพีจีเอ นั้นจะมีผู้ผลิตหลัก 2 ราย ได้แก่บริษัท XILINX และบริษัท Intel (<http://www.intel.com>) ซึ่งได้ซื้อกิจการจากบริษัท Altera ในบทความนี้จะใช้ซอฟต์แวร์ ISE WebPACK ของบริษัท Xilinx สามารถดาวน์โหลดได้ที่เว็บไซต์ <http://www.xilinx.com> ส่วนซอฟต์แวร์สำหรับการจำลองวงจรโค้ด VHDL ใช้ซอฟต์แวร์ ISim ที่มาพร้อมกับโปรแกรม ISE WebPack หรือใช้ ModelSim Student Edition สามารถดาวน์โหลดได้ที่เว็บไซต์ของบริษัท Mentor Graphics (<http://www.mentor.com>) แต่ในตัวอย่างผลการจำลองที่แสดง จะใช้งานโปรแกรม ISim เป็นหลัก

ซอฟต์แวร์สำหรับการออกแบบเอฟพีจีเอ ของบริษัท Xilinx มี 2 เวอร์ชัน ได้แก่

- Vivado Design Suite HLx
- ISE Design Suite

โดยจะมีเวอร์ชันใช้งานฟรี ได้แก่ Vivado HL WebPack หรือ ISE WebPack



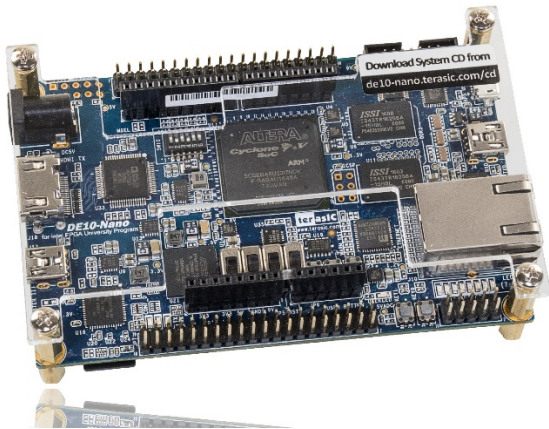
ซอฟต์แวร์สำหรับการออกแบบเอฟพีจีเอ ของบริษัท Intel

- Intel® Quartus® Prime Design ซึ่งจะมีเวอร์ชันฟรีเป็น Quartus Prime Lite Edition

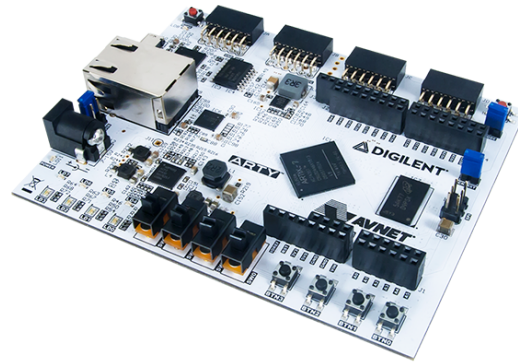


บอร์ดเอฟพีจีเอ

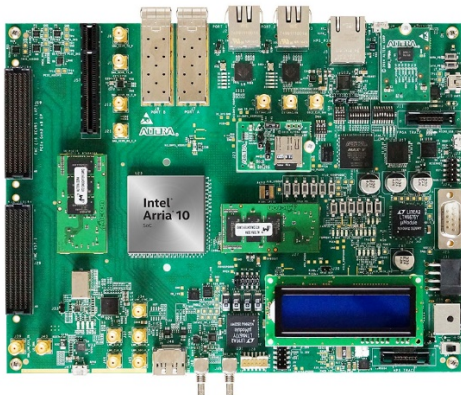
ในการพัฒนางจรด้วยเอฟพีจีเอ นักออกแบบจำเป็นต้องทำการทดสอบหรือพิสูจน์การทำงานของระบบโดยใช้บอร์ดพัฒนาเอฟพีจีเอ ซึ่งบอร์ดพัฒนาจะมีตั้งแต่บอร์ดขนาดเล็ก จำนวนลอจิกเกตไม่มาก จนถึงบอร์ดที่มีขีดความสามารถสูงสำหรับการพัฒนาประยุกต์ใช้งานที่มีความซับซ้อนสูง ราคาบอร์ดจะสูงตามไปด้วย



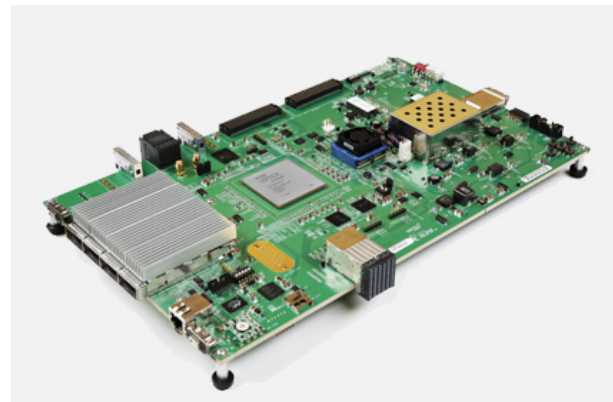
รูปที่ 8 DE-10 nano kit (Terasic)



รูปที่ 9 Arty A7: Artix-7 FPGA Development Board (Digilent Inc.)

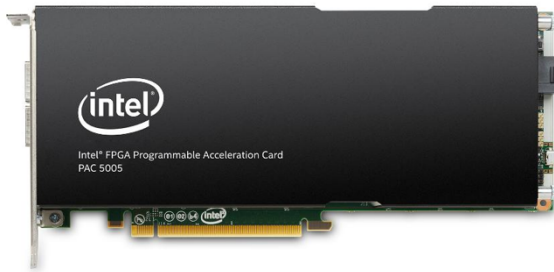


รูปที่ 10 บอร์ด Intel® Arria® 10 SoC Development Kit (Intel)



รูปที่ 11 บอร์ด Xilinx Virtex UltraScale FPGA VCU110 Development Kit (Xilinx)

การ์ดเร่งความเร็ว (Accelerator Card) เป็นการ์ดเอฟพีจีเอ ที่ใช้งานเร่งความเร็วของการทำงานของคอมพิวเตอร์หรือเซิร์ฟเวอร์ เช่นการรับส่งข้อมูลแบนด์วิธสูงในศูนย์ข้อมูล หรือเพิ่มความเร็วในการคำนวณหรือการประมวลผลในคอมพิวเตอร์



รูปที่ 12 การ์ด Intel FPGA PAC D5005 (Intel)



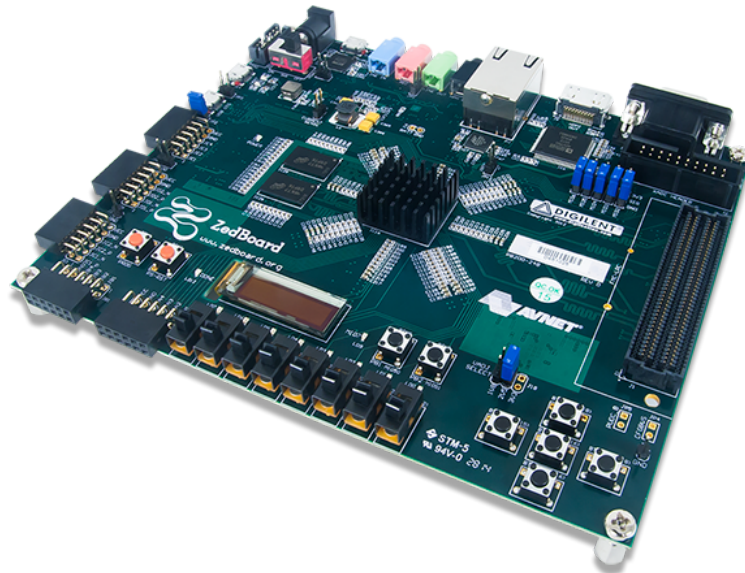
รูปที่ 13 การ์ด Xilinx Alveo U250 (Xilinx Inc.)

การ์ดเร่งการส่งข้อมูลสำหรับ Data Center จากค่าย Intel และ Xilinx นอกจากนี้ยังใช้สำหรับการคำนวณและประมวลผลทางด้าน AI

บอร์ด Digilent ZedBoard Zynq-7000 ARM/FPGA SoC Development Board

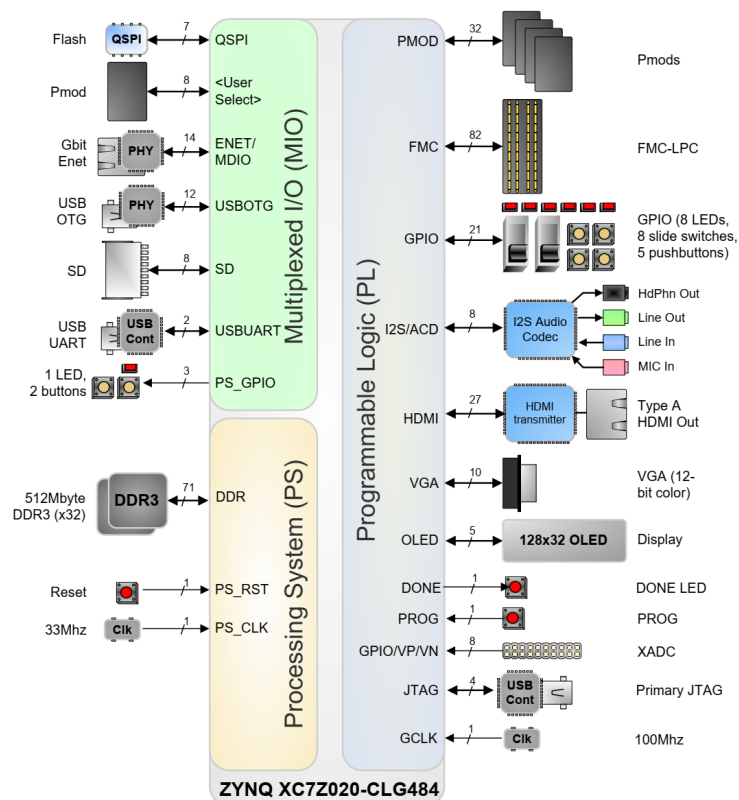
บอร์ดตัวอย่างที่ใช้ประกอบการทดลองเป็นบอร์ดรุ่น ZedBoard Zynq-7000 ARM/FPGA SoC ผลิตโดยบริษัท Digilent Inc. ประเทศสหรัฐอเมริกา (<http://www.digilentinc.com>) ส่วนประกอบของบอร์ดประกอบด้วย

- Xilinx Zynq-7000 AP SoC XC7Z020-CLG484
- Dual-core ARM Cortex-A9
- 512 MB DDR3
- 256 MB Quad-SPI Flash
- On-board USB-JTAG Programming
- 10/100/1000 Ethernet
- USB OTG 2.0 and USB-UART
- Analog Devices ADAU1761 SigmaDSP® Stereo Audio Codec
- Analog Devices ADV7511 High Performance 225 MHz HDMI Transmitter (1080p HDMI, 8-bit VGA, 128x32 OLED)
- PS & PL I/O expansion (FMC, Pmod, XADC)



รูปที่ 14 บอร์ดรุ่น ZedBoard Zynq-7000 (Digilent Inc)

ไดอะแกรมของบอร์ด แสดงดังรูปที่ 15



รูปที่ 15 ไดอะแกรมของบอร์ด (ZedBoard Hardware User's Guide v2.2)

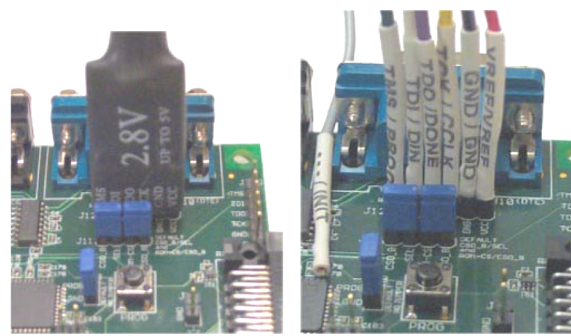
สำหรับรายละเอียดของบอร์ด การกำหนดขา (pin) ต่างๆ วิธีการดาวน์โหลดโปรแกรมลงบอร์ดให้ศึกษาจากคู่มือการใช้งานบอร์ด ตามเอกสารอ้างอิง [5]

การโปรแกรมเอฟพีจีเอ

การโปรแกรมเอฟพีจีเอมีหลายวิธี (โหมด) โดยทั่วไป จะโปรแกรมไปที่เอฟพีจีเอโดยตรง ในกรณีของการทดสอบขณะทำการพัฒนาระบบ หรือลงโปรแกรมในอุปกรณ์เก็บความจำเช่น Flash หรือ EEPROM ซึ่งใช้คู่กับอุปกรณ์เอฟพีจีเอ ในกรณีใช้ผลิตภัณฑ์ในภาคสนาม

การโปรแกรมเอฟพีจีเอ โดยใช้สายดาวน์โหลดผ่าน JTAG

การโปรแกรมหรือดาวน์โหลดข้อมูลลงบนชิพเอฟพีจีเอในโหมดของ JTAG mode ซึ่งสามารถทำการต่อสาย JTAG หรือต่อผ่านสาย USB กับบอร์ด FPGA ได้ดังรูปที่ 16

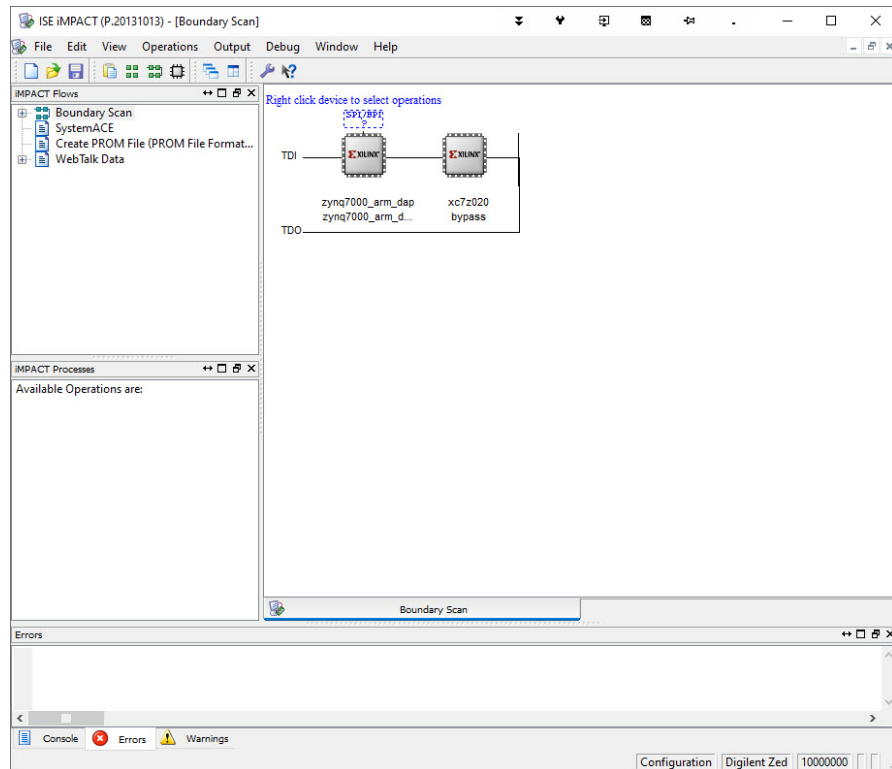


JTAG Parallel Connector & Parallel Cable with Flying Leads

รูปที่ 16 การเชื่อมต่อสาย JTAG

การโปรแกรมเอฟพีจีเอโดยใช้โปรแกรม iMPACT ผ่านสาย USB

เราสามารถทำการโปรแกรมชิพเอฟพีจีเอได้โดยตรงจากหน้าต่างโปรแกรม Xilinx ISE Project Navigator โดยการรันโปรแกรม iMPACT ดังรูปที่ 17 โดยโปรแกรมจะทำการสแกนหาอุปกรณ์ไอซีเอฟพีจีเอ ที่เชื่อมต่ออยู่

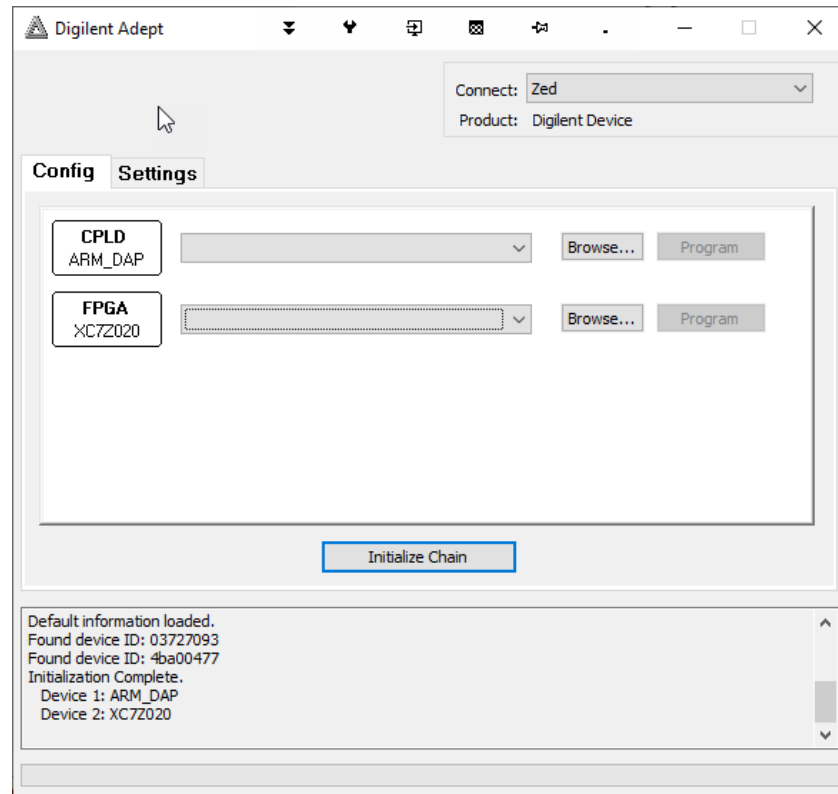


รูปที่ 17 การโปรแกรมเอฟพีจีเอผ่านโปรแกรม iMPACT

การโปรแกรมเอฟพีจีเอโดยใช้ซอฟต์แวร์ Adept2.x ผ่านสาย USB

สาย USB นั้นนอกจากจะใช้เป็นแหล่งจ่ายไฟเลี้ยงบอร์ดแล้ว ยังสามารถใช้สำหรับการโปรแกรมเอฟพีจีเอ โดยผ่านซอฟต์แวร์ Adept

ทำการดาวน์โหลดและติดตั้งโปรแกรมซอฟต์แวร์ Adept โดยดาวน์โหลดซอฟต์แวร์จากเว็บไซต์ Digilent Inc (<http://www.digilentinc.com>) ก่อนติดตั้งซอฟต์แวร์ ต้องแน่ใจว่าบอร์ดไม่ได้เชื่อมต่ออยู่กับเครื่องคอมพิวเตอร์ การติดตั้งทำได้โดยการรันโปรแกรมที่ดาวน์โหลดมาและให้ปฏิบัติตามขั้นตอนต่างๆ จนกระทั่งติดตั้งไดรเวอร์ของบอร์ด เมื่อเสร็จเรียบร้อยแล้ว โปรแกรมจะสามารถเปิดได้จาก Shortcut ใน Start menu ในขั้นตอนการโปรแกรม ให้ทำการเชื่อมต่อบอร์ด ผ่านทางสาย USB หลังจากนั้นให้เริ่มโปรแกรม Adept โดยการคลิก Start → Digilent → Adept



รูปที่ 18 การโปรแกรมลงบนบอร์ดผ่านทางสาย USB โดยใช้ซอฟต์แวร์ Adept เมื่อเชื่อมต่อกับบอร์ด ZedBoard Zynq-7000

ตัวอย่างโค้ดภาษา VHDL วงจร FF (Flip-flop)

ภาษา VHDL เป็นภาษาระดับสูงใช้สำหรับการออกแบบระบบดิจิทัล มีความยืดหยุ่นสูง รูปที่ 19 แสดงตัวอย่างโค้ดที่ใช้บรรยายวงจร Flip-flop เมื่อนำโค้ดที่ได้ไปทำการสังเคราะห์วงจร จะได้วงจรตามรูปที่ 20

```

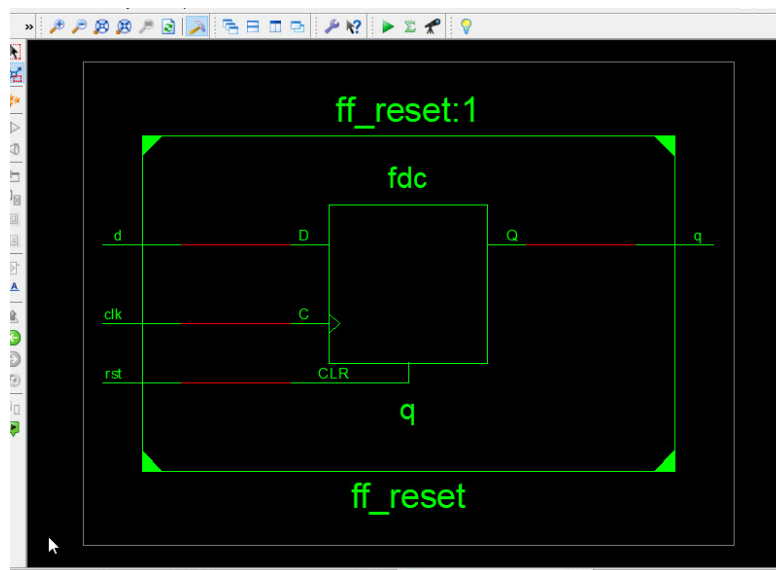
1  library ieee;
2  use ieee.std_logic_1164.all;
3  entity ff_reset is
4  port (
5      d      : in  std_logic; -- data input
6      clk     : in  std_logic; -- clock
7      rst     : in  std_logic; -- reset, active hi
8      q      : out std_logic); -- output
9  end entity ff_reset;
10
11 architecture rtl of ff_reset is
12
13 begin -- architecture rtl
14
15     process (clk, rst) is
16     begin -- process
17         if (rst = '1') then
18             q <= '0';
19         elsif (clk'event and clk = '1') then
20             q <= d;
21         end if;
22     end process;
23 end architecture rtl;
24

```

I

รูปที่ 19 ตัวอย่างโค้ดภาษา VHDL

ผลของการสังเคราะห์วงจร ดังรูปที่ 20



รูปที่ 20 วงจร ff_reset

จากที่กล่าวมา จะเห็นว่าในการพัฒนางานด้าน IoT โดยใช้เอพฟิจีเอ จำเป็นต้องใช้ทักษะและเครื่องมือหลายอย่าง ประกอบกัน ในส่วนของผู้ผลิตอุปกรณ์ได้มีการพัฒนาเครื่องมือใหม่ช่วยในการประยุกต์ใช้เอพฟิจีเอสำหรับผู้ที่ไม่

เชี่ยวชาญทางด้านฮาร์ดแวร์ อย่างเช่น OpenVINO toolkit (<https://software.intel.com/en-us/openvino-toolkit>) หรือ PYNQ (<http://www.pynq.io/>) เพื่อให้โปรแกรมเมอร์สามารถใช้งานเอฟพีจีเอได้สะดวกมากขึ้น

เอกสารอ้างอิง

- [1] Venuto D.D., Annese V.F., Sangiovanni-Vincentelli A.L. The ultimate IoT application: A cyber-physical system for ambient assisted living. Proceedings of the 2016 IEEE International Symposium on Circuits and Systems (ISCAS); 2016 May 22–25; Montreal, Canada. pp. 2042–45
- [2] Why Microsoft Is Betting on FPGAs for Machine Learning at the Edge. [Internet]. c2019 [cited 2020 Jan 06]. Available from: <https://www.datacenterknowledge.com/edge-computing/why-microsoft-betting-fpgas-machine-learning-edge>
- [3] ชำนาญ ปัญญาใส และวัชรกร หนูทอง. ภาษา VHDL สำหรับการออกแบบวงจรดิจิทัล. กรุงเทพฯ: สำนักพิมพ์ซีไอเคยูเคชั่น; 2547.
- [4] All About FPGAs [อินเทอร์เน็ต]. 2549 [เข้าถึงเมื่อ 6 ม.ค. 2563]. เข้าถึงได้จาก: <https://www.eetimes.com/all-about-fpgas/>
- [5] ZedBoard Zynq-7000 Development Board Reference Manual [อินเทอร์เน็ต]. [เข้าถึงเมื่อ 6 ม.ค. 2563]. เข้าถึงได้จาก: <https://reference.digilentinc.com/reference/programmable-logic/zedboard/reference-manual>
- [6] ZedBoard Hardware User's Guide v2.2 [อินเทอร์เน็ต]. 2557. [เข้าถึงเมื่อ 6 ม.ค. 2563]. เข้าถึงได้จาก: http://zedboard.org/sites/default/files/documentations/ZedBoard_HW_UG_v2_2.pdf