

PEMANFAATAN *FIELD-PROGRAMMABLE GATE ARRAYS (FPGAS)* DALAM PENGEMBANGAN OTOMASI KENDALI DAN INDUSTRI

Irfan Darmawan¹⁾, Asep Andang¹⁾

1) Jurusan Teknik Elektro Universitas Siliwangi

Sistem Instrumentasi dan Kontrol

Sistem instrumentasi dan kendali dapat dilukiskan tersusun atas beberapa komponen : (1) sensor (sering juga dinamakan transduser) adalah system yang mengukur besaran lingkungan (misalnya suhu, tekanan, kelembaban, dsb.) dan mengkonversikannya menjadi besaran listrik (2) analog-to-digital converter adalah system yang mengkonversikan besaran listrik analog menjadi besaran listrik digital (3) *information processing* adalah system yang memproses besaran digital menjadi besaran digital yang lain (4) display adalah system yang menampilkan suatu besaran digital, misalnya layar monitor, LCD display, 7-segment display, dsb. (5) digital-to-analog converter adalah sistem yang mengkonversikan besaran listrik digital menjadi besaran listrik analog (6) actuator adalah system yang mengkonversikan besaran listrik analog menjadi besaran lainnya misalnya kecepatan putaran.

FPGAs dan Programmable Logic Devices

Field-Programmable Gate Arrays (FPGAs) adalah salah satu perangkat yg termasuk dalam kelompok programmable logic devices. FPGAs berbeda dari general-purpose mikroprosesor (misalnya Intel) dalam hal fleksibilitas logic-nya. Mikroprosesor mempunyai hardware yang

Tetap. Assembly programmer memprogram suatu komputasi dengan keterbatasan pada tetapnya banyaknya register, siklus fetch-decode-execute, serta fungsi-fungsi ALU(arithmetic and logic unit) dan pada banyaknya bit suatu register. FPGAs berbeda dari mikrokontroler (misalnya ATMEL), karena mikrokontroler pada prinsipnya adalah mikroprosesor yang diprogram dengan bahasa assembly dan dirancang sebagai pengendali bukan untuk komputasi. Mikroprosesor dan mikrokontroler mengimplementasikan suatu komputasi pada hardware yang tetap. Hardware pada FPGAs diserahkan sepenuhnya pada design engineer untuk memprogramnya. Sebelum diprogram, FPGAs hanyalah tersusun atas blok-blok yang belum dikonfigurasi dan interkoneksi yang belum disusun dan difungsikan. Oleh karena itu, istilah yang lebih tepat adalah merekonfigurasi FPGAs, bukan memprogramnya. Chip FPGAs yang sama dikonfigurasi dengan data yang berbeda akan mengimplementasikan hardware yang berbeda.

FPGAs berbeda dari PLAs (Programmable Logic Arrays) maupun PALs(Programmable Array Logic). PLAs dan PALs tersusun atas matrix gerbang OR dan matrix gerbang AND. PLAs mengimplementasikan bentuk sum-of-product (SOP) dari logika kombinasi,

sedangkan PALs mengimplementasikan bentuk product-of-sum (POS) dari logika kombinasi. PALs dan PLAs hanya dapat mengimplementasikan rangkaian kombinasi, FPGAs dapat mengimplementasikan rangkaian kombinasi maupun rangkaian sekuensial.

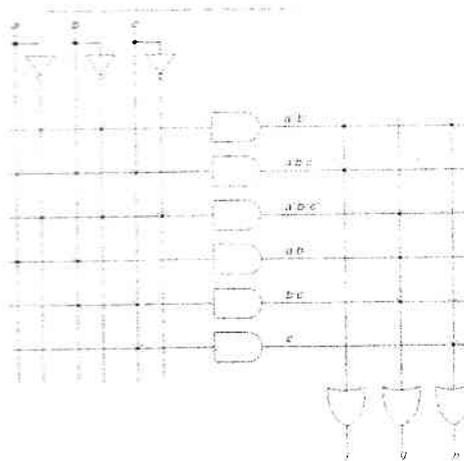
Rangkaian kombinasi dapat diimplementasikan menggunakan ROM (Random-Access Memory). ROM8x1 dapat mengimplementasikan sembarang fungsi Boole yang mempunyai sebuah output 1-bit dan mempunyai 3-input masing-masing 1-bit. Banyaknya variasi fungsi Boole yang dapat diimplementasikan dalam sebuah ROM8xi adalah $2^{2^3} = 256$ buah. ROM juga termasuk programmable logic device tetapi khusus untuk rangkaian kombinasi. Kelemahan ROM adalah tingginya pemborosan silikon bila banyak isinya adalah logic 0. Misalnya bila diimplementasikan $F = A.B.C$ pada ROM8x1, maka semua word pada ROM bernilai 0 kecuali pada address 7 (111 biner). PLAs dikembangkan untuk mengatasi pemborosan pada ROM ini. Namun PLAs mempunyai keterbatasan dibandingkan ROM.

Bila pada ROMkx1 sembarang fungsi Boole yang mempunyai sebuah output 1-bit dan mempunyai $\log_2(k)$ -input masing-masing 1-bit dapat implemntasikan, maka pada PLAs hanya fungsi yang dapat diungkapkan dalam bentuk SOP menggunakan maksimum p suku saja yang dapat diimplementasikan. Suatu PLAs nxm dangan p suku hasil-kali tersusun atas p gerbang AND masing-masing 2n input dan m gerbang OR masing-masing p-input.

Tabel 1 ROM8x1 mengimplementasikan $F=A.B.C$

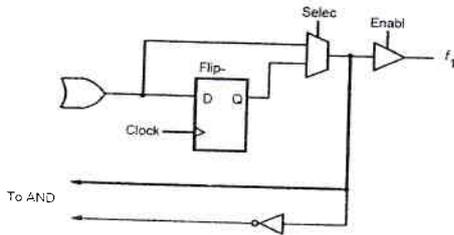
Address (desimal)	Address (biner)	Data ROM8x1
0	000	0
1	001	0
2	010	0
3	011	0
4	100	0
5	101	0
6	110	0
7	111	1

Gambar 1. melukiskan implementasi dari fungsi-fungsi $f(a,b,c) = a'b' + abc$, $g(a,b,c) = a'b'c' + ab + bc$, dan $h(a,b,c) = c$, pada PLA. Terlihat bagaimana kaki-kaki disusun dengan menghubungkan matrix AND, dan bentuk SOP disusun dengan menghubungkan matrix OR.



Gambar 1. Implementasi fungsi-fungsi $f(a,b,c) = a'b' + abc$, $g(a,b,c) = a'b'c' + ab + bc$, dan $h(a,b,c) = c$, pada PLA.

PLA dan PAL mempunyai keterbatasan dalam ketidakmampuannya mengimplementasikan rangkaian sekuensial. Oleh karena itu ditemukanlah CPLD (*Complex Programmable Logic Device*). Gambar 2 melukiskan arsitektur CPLD.

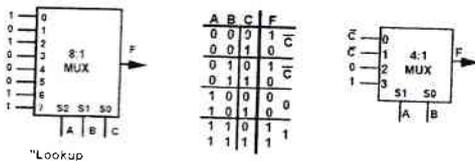


Gambar 2. Complex Programmable Logic Device Block

Multiplexer Sebagai General Purpose Logic Block

Andaikan akan diimplementasikan $F(A,B,C) = m_0 + m_2 + m_6 + m_7 = A'B'C' + A'B'C + AB'C + ABC = A'B'(C') + A'B(C') + A'B(0) + AB(1)$.

Tabel kebenaran, implementasi dalam MUX8-ke-1, dan implementasi dalam MUX4-ke-1 ditunjukkan pada Gambar 3. Terlihat bahwa MUX dapat mengimplementasikan sembarang fungsi kombinasi. Karena F merupakan fungsi dari 3 variabel, maka bila hanya mempunyai MUX4-ke-1 harus dilakukan functional decomposition dahulu.

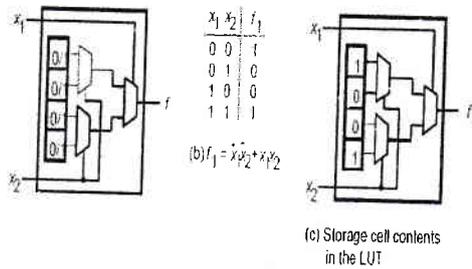


Gambar 3. Implementasi $F(A,B,C) = m_0 + m_2 + m_6 + m_7$ dalam MUX8-ke-1 dan MUX4-ke-1

FPGAs merupakan komponen utama suatu reconfigurable hardware. Reconfigurable teknologi didukung oleh FPGAs dan Digital Signal Processor.

Komponen Dasar FPGAs

FPGAs mempunyai komponen dasar yang dinamakan LUT (Look-Up Table) untuk mengimplementasikan rangkaian kombinasi, register (flip-flop) untuk mengimplementasikan elemen penyimpan/rangkaian sekuensial, carry logic untuk fungsi aritmatika, dan expansion logic untuk mengimplementasikan fungsi yang tergantung pada lebih dari 4 input. Gambar 4 melukiskan LUT dengan dua buah input x_1 dan x_2 mengimplementasikan fungsi $f_1 = x_1 \text{ xnor } x_2$.

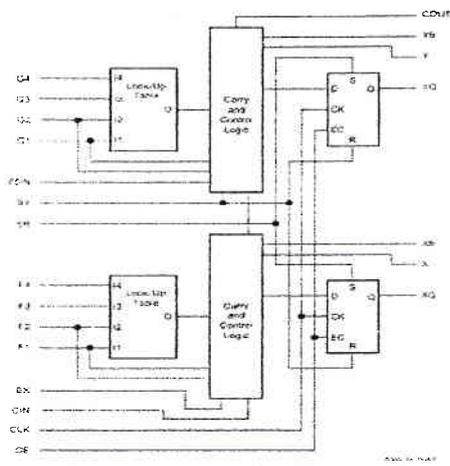


Gambar 4. LUT 2-input x_1 dan x_2 mengimplementasikan fungsi $f_1 = x_1 \text{ xnor } x_2$.

Gambar 4 melukiskan 2 buah LUT 4-input pada sebuah CLB (Combinational Logic Block) dari Xilinx Spartan IIE. Pada gambar 12 terlihat bahwa CLB selain memiliki dua buah LUT, juga ada dua buah fast carry logic dan dua buah D flip-flop.

Keberadaan flip-flop membuat CLB juga mampu mengimplementasikan rangkaian sekuensial. D Flip-flop yang bersifat edge-triggered juga cocok untuk implementasi

rangkaian sekuensial dengan metoda penyandian one-hot encoding yang walaupun memerlukan lebih banyak flip-flop dibandingkan binary encoding, tetapi next-state logic dan output logic-nya lebih sederhana sehingga delay lebih rendah dan bisa diclock dengan frekuensi yang lebih tinggi.

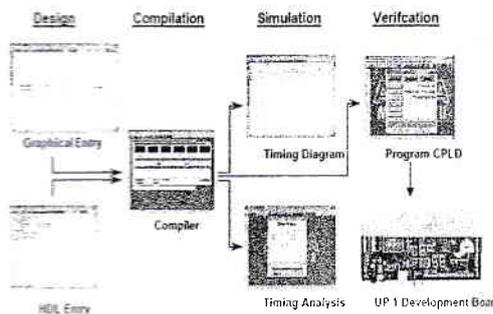


Gambar 5. Teknologi Rekonfigurasi FPGAs

Langkah-langkah merekonfigurasi FPGAs menggunakan contoh Altera digambarkan pada Gambar 5. Design engineer dapat memilih apakah menggunakan schematic capture (graphical entry) ataukah menggunakan HDL(hardware description language) entry. HDL entry mempunyai banyak keunggulan dibandingkan graphical entry, di antaranya :

1. Pada HDL entry, modifikasi ukuran / banyaknya bit suatu register/bus jauh lebih mudah dibandingkan dengan graphical entry, karena pada graphical entry akan mengharuskan rancangan digambar ulang

2. Design engineer pada HDL entry dapat memilih deskripsi behavioral yang memungkinkan struktur hardware yang dirancang ditemukan sepenuhnya oleh software. Engineer cukup menuliskan deskripsi wataknya bukan bentuknya.
3. HDL entry lebih bersifat portabel karena bentuknya text, sehingga dapat dikompile dengan tool yang berlainan. Sementara itu untuk graphical entry, banyak tool berbeda format.
4. HDL entry lebih cocok untuk design team yang tersusun dari banyak engineer, karena lebih modular, dan lebih mudah mengkomunikasikan design antar engineers.



Gambar 6. Langkah-langkah Perancangan FPGAs

Selanjutnya rancangan dikompilasikan. Yaitu dilakukan DRC (*Design Rule Check*), dan bila ada error akan dilaporkan. Design engineer kemudian dapat membetulkan error yang terjadi. Bila DRC tidak lagi memberikan pesan kesalahan, design engineer dapat mencek watak circuit yang akan dihasilkannya menggunakan timing diagram.

Design engineer juga dapat menguji clocking scheme dengan melakukan timing analysis untuk

mengetahui delay-delay pada circuit dan berapa maksimum frekuensi clock. Setelah itu rancangan dapat diubah menjadi configuration bit-stream dan diload ke dalam device sesungguhnya.

Salah satu bahasa untuk HDL entry adalah VHDL (*VHSIC Hardware description Language* ; *VHSIC = Very High Speed Integrated Circuit*).

Keterbatasan FPGAs

FPGAs tidak dapat menggantikan mikroprosesor sepenuhnya, sepertihalnya Wireless LAN (IEEE 802.11) tidak akan menggantikan Wired LAN pada dunia computer network. FPGAs bersifat komplementer dengan mikroprosesor.

Otomasi, kendali, dan robotika mungkin menggunakan algoritma tertentu, misalnya jaringan syaraf buatan (ANN = artificial neural network). Pada ANN diperlukan adanya representasi floating-point untuk bobot koneksi ke suatu neuron. Misalnya IEEE Floating-point format 32-bit tersusun atas 1-bit tanda(S), 8-bit eksponen (E) dalam sandi excess-127 dan 23-bit mantissa (M) dalam bentuk ternormalisir. Kemudian bobot neuron mesti dikenakan suatu operasi perkalian. Kalau diimplementasikan multiplier dalam hardware tentu diperlukan sumberdaya yang besar (perkalian 2 buah bilangan 32-bit memberikan maksimum 64-bit hasil perkalian). Kalau perkalian diimplementasikan dengan algoritma Booth, sumberdaya yang diperlukan berkurang tetapi kinerja turun.

Sementara itu, banyaknya neuron masih ditentukan pula oleh banyaknya hidden layer, demikian pula banyaknya koneksi dan bobot yang memasuki suatu neuron. Sehingga masih belum

dimungkinkan mengimplementasikan ANN dalam sebuah FPGAs.

Implementasi dalam banyak FPGAs masih merupakan persoalan terutama dalam hal pemecahan (partisi) desain ke dalam beberapa FPGAs. Kalau partisi manual, mungkin banyak peluang optimasi yang terlewatkan. Kalau partisi otomatis, saat ini masih sulit dilakukan. Andrew, Niehaus, dan Ashenden (2004) mengindikasikan diperlukannya suatu programming model yang sesuai untuk implementasi hybrid CPU/FPGA.

FPGAs saat ini cocok untuk akselerator dari suatu bagian komputasi yang paling menyita sumberdaya komputasi. Bila bagian yang akan di-FPGA-kan kita pilih dengan baik, kinerja secara keseluruhan akan meningkat. Bidang otomasi, kendali, dan robotika akan menarik manfaat dari ketersediaan berbagai alternatif implementasi : mikroprosesor, mikrokontroler, PLC, FPGAs, BASIC-Stamp processor, dsb. Kurang bijaksana untuk sepenuhnya memikulkan seluruh beban kepada satu teknologi saja. Aspek yang terkait sangat banyak dan domain-nya sangat luas, sepertihalnya bidang ICT (Information and Communication Technology) itu sendiri bukan hanya berada di pundak ilmu komputer/informatika, elektronika, teknik elektro, fisika, matematika, dsb. tetapi pada semua bidang ilmu yang mungkin ada.

Kesimpulan

FPGA sebagai suatu platform untuk implementasi sistem digital merupakan alternative yang menarik dan berpotensi menjadi komponen utama masa depan, dan dengan dikembangkannya SoC maka kita dapat merancang system jumlah sesuai dengan keinginan dan waktu yang efisien

Daftar Pustaka

Andrews, D., D. Niehaus, dan P. Ashenden, 2004: *Programming Models for Hybrid CPU/FPGA Chips*, IEEE Computer pp 118-120. January

Sofyan, dan Jazi Eko Istiyanto, 2005 : *Pembuatan Prototipe Mesin Pelubang PCB Berbasis Komputer dan Optimasi Kinerjanya Menggunakan Algoritma NNH*, SNIKT12005, Universitas Kristen Satya Wacana, Salatiga (diterima untuk publikasi).

<http://www.kressary.de>.