

Briefing

Pada bab briefing ini, praktikan akan mempelajari tentang sejarah awal FPGA, konsep dasar dan cara kerja FPGA, konsep dasar gerbang logika, serta software dan hardware yang akan digunakan dalam praktikum FPGA. Asisten praktikum atau praktikan dapat membaca tujuan dan persyaratan praktikum bab ini agar praktikum dapat berjalan sesuai prosedur.

Tujuan

Tujuan	Penjelasan
Memahami sejarah perkembangan FPGA	Praktikan diharapkan dapat memahami sejarah awal perkembangan FPGA dan bagaimana teknologi ini berevolusi hingga digunakan secara luas saat ini
Memahami konsep dasar dan cara kerja FPGA	Praktikan diharapkan dapat memahami komponen FPGA dan cara kerja pada FPGA
Memahami jenis-jenis software dan hardware FPGA	Praktikan diharapkan dapat memahami software dan hardware yang akan digunakan untuk desain dan simulasi FPGA
Memahami konsep dasar dari gerbang logika	Praktikan diharapkan dapat mengenali dan menjelaskan fungsi dasar gerbang logika seperti AND, OR, NOT,NAND,NOR,XOR,XNOR dan bagaimana gerbang-gerbang ini digunakan dalam desain FPGA.

Persyaratan

Disarankan praktikan menggunakan hardware dan software sesuai pada dokumentasi ini. Apabila terdapat versi hardware atau software yang cukup lama dari versi yang direkomendasikan maka sebaiknya bertanya kepada Asisten Mengajar Shift.

HARDWARE YANG DIBUTUHKAN PRAKTIKUM

PC / Laptop

FPGA Board Nexys A7 dan Nexys 4

Kabel Power USB

SOFTWARE YANG DIBUTUHKAN PRAKTIKUM

Vivado Design Suite



Diusahakan untuk memakai **Versi** dan **Aplikasi** yang sama agar tidak terjadinya kesalahan yang tidak diinginkan!

B.1. Sejarah FPGA

FPGA (*Field Programmable Gate Array*) merupakan sebuah IC digital yang digunakan untuk merangkai rangkaian digital. FPGA ini mulai diperkenalkan pada tahun 1980 dan dikembangkan sejak tahun 1985 oleh perusahaan Xilinx yang berbasis di San Jose, California yang menciptakan FPGA pertama adalah XC2064 pada tahun 1985. Perangkat ini sangat primitif, dengan hanya 800 gerbang, hanya sebagian kecil jika dibandingkan dengan jutaan operasi gerbang yang dapat dilakukan pada FPGA saat ini. Perangkat ini juga relatif mahal, yaitu \$55, yang jika disesuaikan dengan inflasi akan menjadi sekitar \$145 saat ini. Meskipun demikian, XC2064 telah memulai seluruh industri, dan Xilinx telah menjadi salah satu perusahaan dominan di pasar FPGA selama lebih dari 30 tahun. Selain Xilinx, terdapat juga beberapa perusahaan lain yang memproduksi FPGA seperti Altera, Lattice, dan Quicklogic.

B.1.1 Versi Produk FPGA Xilinx

1. Virtex Family
2. Kintex Family
3. Artix Family
4. Zynq Family
5. Spartan Family
6. Easy Path
7. Versal



Gambar B. 1 Nexys A7

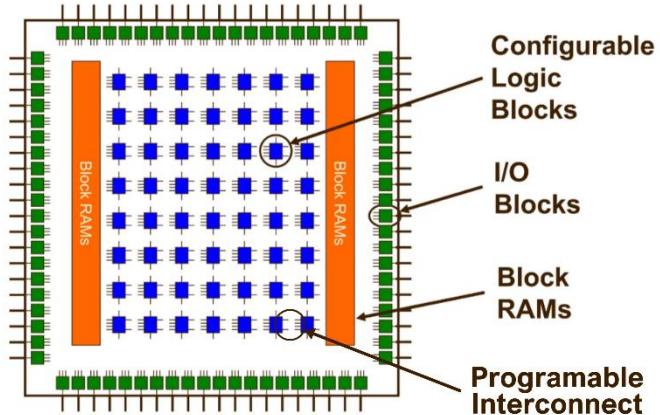


Gambar B. 2 Nexys 4

B.2. Komponen FPGA

Bila dilihat dari namanya, (*Field Programmable*) dapat diartikan bahwa FPGA ini bersifat dapat dirancang sesuai dengan keinginan dan kebutuhan user atau pemakai.

Sedangkan (*Gate Array*) artinya FPGA ini terdiri atas gerbang-gerbang digital dimana masing-masing dari gerbang tersebut dapat dikonfigurasikan antara satu sama lain. FPGA ini juga memiliki sifat *volatile*, yaitu sifat dimana apabila sumber dayanya dicabut maka program yang sudah ditanam sebelumnya akan hilang. Namun dibeberapa perangkat, sudah ada yang ditanamkan ROM untuk dapat menyimpan program yang sudah dibuat.



Gambar B. 3 Komponen FPGA

Dalam FPGA terdapat komponen-komponen yang memiliki fungsinya masing-masing yaitu :

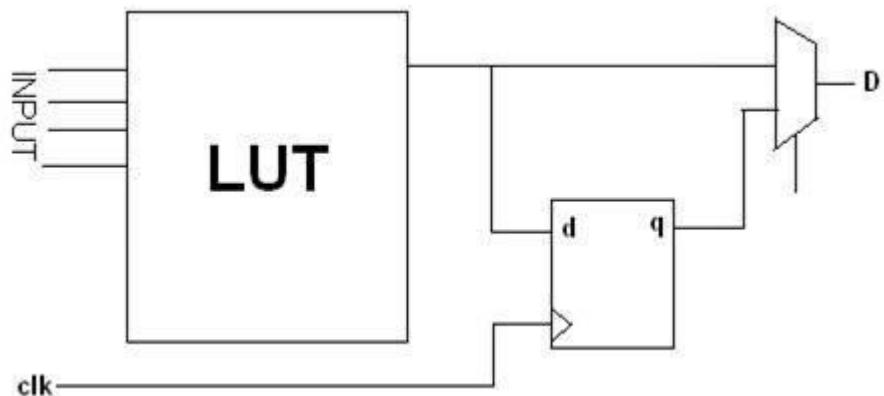
- *Configure Logic Blocks* (CLB), adalah komponen utama dari FPGA yang berfungsi memproses segala bentuk logika yang di blok user atau pengguna. CLB biasanya terdiri dari beberapa elemen logika (gerbang logika, LUT (*Look Up Tables*), Flip-Flop, dan multiplexer) dan digunakan untuk memproses rangkaian logika yang dibuat oleh pengguna.
- I/O Blocks, berfungsi sebagai *interface* yang dapat diprogram untuk *input* dan *output* transfer data dalam FPGA.
- Block RAM, blok memori yang berfungsi sebagai penyimpanan data dalam FPGA

Programmable Interconnect, bagian ini berfungsi sebagai saklar yang menghubungkan antara satu CLB dengan CLB lainnya.

B.3. Cara Kerja FPGA

FPGA tersusun dari *logic block* yang saling terhubung satu sama lain. Kumpulan-kumpulan dari *logic block* ini berjumlah ratusan bahkan ribuan sehingga membentuk suatu

fungsi yang kompleks. Sebuah *logic block* pada dasarnya terdiri dari sebuah LUT (*Lookup Table*), *Flip-Flop*, dan sebuah *multiplexer*.



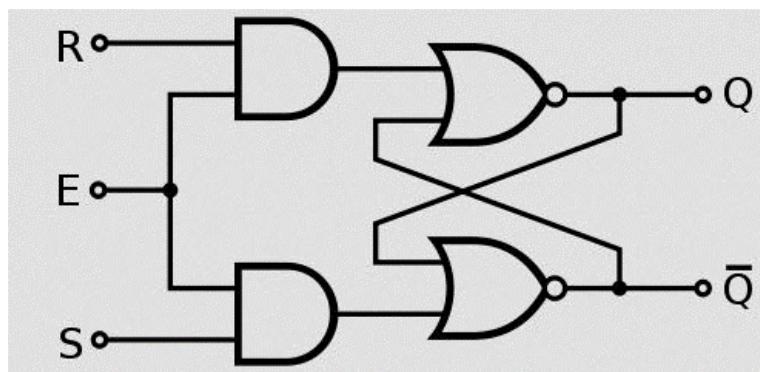
Gambar B. 4 Logic-cell

B.3.1. LUT (Lookup Table)

LUT atau *Lookup Table* merupakan sejenis RAM yang berkapasitas kecil. Di dalam FPGA, LUT ini memegang peran dalam proses implementasi fungsi-fungsi logika dan mempunyai ciri khas yaitu memiliki 4 masukan.

B.3.2. Flip-Flop

Flip-flop merupakan sebuah rangkaian yang memiliki dua keadaan stabil dan mewakili satu bit. *Flip-flop* adalah *resource* penyimpanan terkecil pada FPGA. Setiap *flip-flop* dalam sebuah CLB adalah register biner yang digunakan untuk menyimpan status logika antara siklus clock pada rangkaian FPGA.

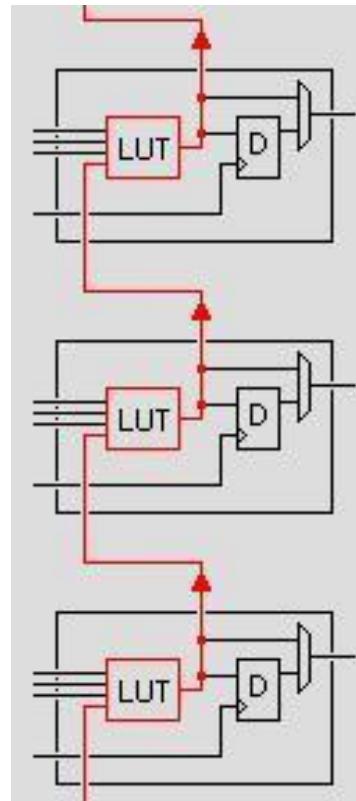


Gambar B. 5 Rangkaian Flip-Flop

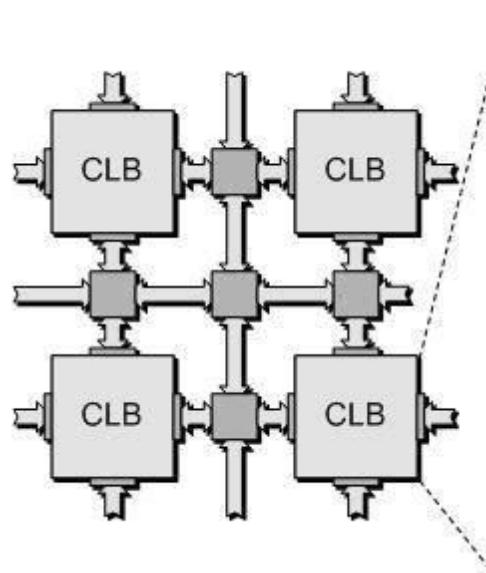
B.3.3. Multiplexer

Multiplexer bekerja sebagai *switch* (saklar) yang menghubungkan beberapa *input* menjadi satu *input-an* saja.

Setiap *logic block* dapat dihubungkan dengan *logic block* lainnya melalui jalur koneksi yang ada. Setiap *block* hanya mampu bekerja secara ringkas dan sederhana, namun apabila antar *block* saling terhubung satu sama lain, maka sebuah fungsi-fungsi logika yang kompleks dapat terbentuk.



Gambar B. 6 Kumpulan Logic-cell



Gambar B. 7 Kumpulan CLB

B.4. Software

Pada umumnya, perusahaan pembuat FPGA memberikan perangkat lunak secara gratis yang sudah termasuk kedalam paket pembelian produk mereka. Namun, software yang gratis ini hanya untuk jenis FPGA tingkat rendah-menengah saja, tetapi sudah cukup mendukung untuk digunakan dalam pembelajaran. Berikut beberapa software pendukung yang gratis :

1. Xilinx, terkenal dengan software miliknya yang bernama ISE Design Suite dan Vivado.

2. Altera, terkenal dengan software miliknya yang bernama Quartus.

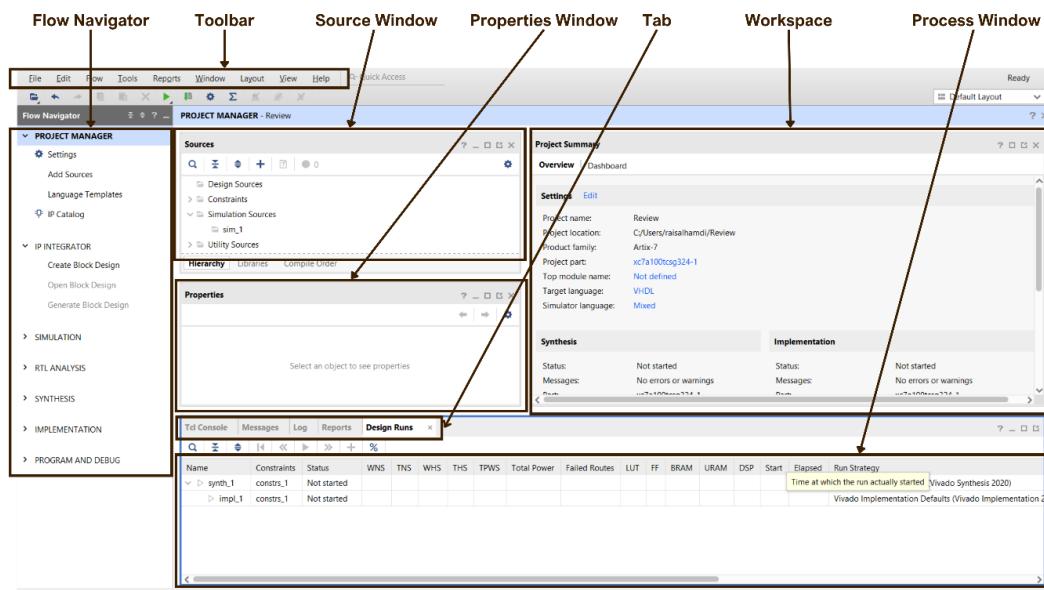
Xilinx, Inc. adalah sebuah perusahaan teknologi asal Amerika yang terutama memasok perangkat logika terprogram. Perusahaan ini merupakan pencipta FPGA. Xilinx juga merupakan perusahaan semikonduktor yang menciptakan model manufaktur niffabrikasi pertama. Didirikan bersama-sama oleh Ross Freeman, Bernard Vonderschmitt, dan James V Barnett II pada tahun 1984, perusahaan ini resmi melantai di NASDAQ pada tahun 1989. Pada bulan Oktober 2020, AMD mengumumkan akuisisi terhadap Xilinx. Akuisisi ini selesai pada tanggal 14 Februari 2022.



Gambar B. 8 Logo Vivado

Pembelajaran praktikum sebelumnya menggunakan *software* ISE Design Suite yang kini diganti menjadi Vivado. Vivado merupakan generasi penerus dari ISE Design Suite, menawarkan sejumlah keunggulan yang membuatnya lebih cocok untuk pembelajaran FPGA saat ini terlebih antarmuka pengguna yang lebih modern dan *user-friendly* dibandingkan dengan ISE Design Suite.

Vivado mendukung sepenuhnya kedua bahasa pemrograman HDL (*Hardware Description Language*) yang paling umum digunakan, yaitu VHDL dan Verilog. Ini memungkinkan pengguna untuk mempelajari dan mengimplementasikan desain FPGA menggunakan bahasa yang mereka kuasai atau yang paling sesuai dengan kebutuhan proyek. *Software* ini dilengkapi dengan alat pengecekan sintaks dan semantik yang kuat. Ini membantu pengguna mengidentifikasi kesalahan dalam kode sebelum proses sintesis, sehingga mempercepat proses debugging dan meningkatkan produktivitas. Hierarki desain dari Vivado membuat pengguna untuk mengorganisasi kode menjadi modul-modul yang lebih kecil dan lebih mudah dikelola.



Gambar B. 9 Tampilan Awal Software Vivado

B.4.1 Proses Status

- Running

Ikon ini menunjukkan bahwa proses sedang berjalan.

- Up-To-Date

Ikon ini menunjukkan bahwa proses berhasil berjalan dengan tidak ada kesalahan atau peringatan dan tidak perlu mengulangi. Jika ikon di samping proses laporan, laporan yang *up-to-date*, namun tugas-tugas terkait dapat memiliki peringatan atau kesalahan. Jika hal ini terjadi, pengguna dapat membaca laporan tersebut untuk menentukan penyebab dari peringatan atau kesalahan.

- Warning Reported

Ikon ini menunjukkan bahwa proses itu berjalan berhasil tetapi ada peringatan yang muncul.

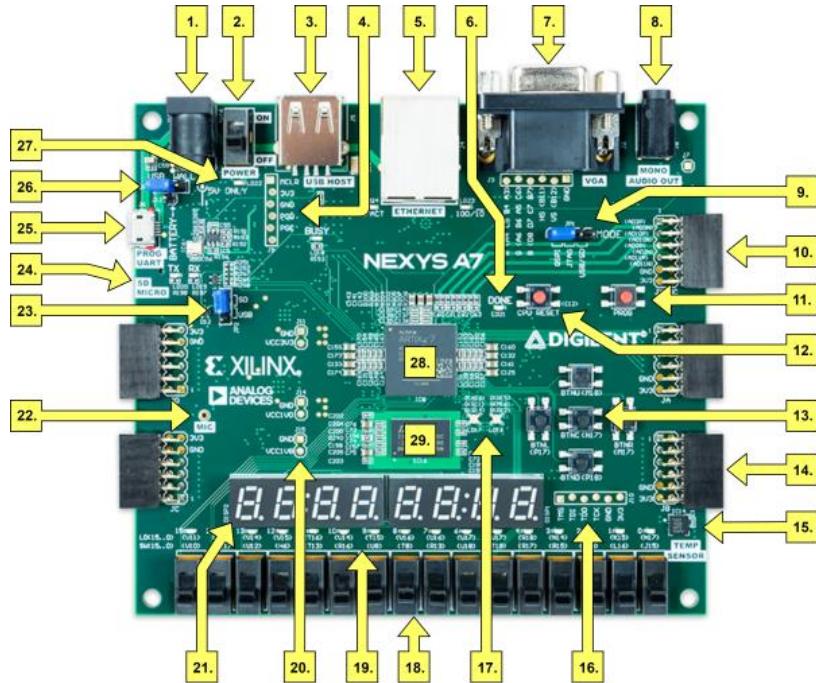
- Error Reported

Ikon ini menunjukkan bahwa proses itu berjalan, tetapi mengalami *error* dan harus di perbaiki.

- Out-Of-Date

Ikon ini menunjukkan bahwa pengguna membuat perubahan desain, yang mengharuskan proses dijalankan ulang. Jika ikon ini berada di samping proses laporan, pengguna dapat menjalankan kembali proses tugas yang terkait untuk membuat versi *up-to-date* laporan.

B.5. Hardware



Gambar B. 10 Pin Callout Nexys A7

Tabel B. 1 Pin Callout dan Keterangannya

Callout	Component Description	Callout	Component Description
1	Power jack	16	JTAG port for (optional) external cable
2	Power switch	17	Tri-color (RGB) LEDs
3	USB host connector	18	Slide switches
4	PIC24 programming port (factory use)	19	LEDs
5	Ethernet connector	20	Power supply test point(s)
6	FPGA programming done LED	21	Eight digit 7-seg display
7	VGA connector	22	Microphone
8	Audio connector	23	External configuration jumper (SD / USB)
9	Programming mode jumper	24	MicroSD card slot
10	Analog signal Pmod port (XADC)	25	Shared UART/ JTAG USB port
11	FPGA configuration reset button	26	Power select jumper and battery header
12	CPU reset button (for soft cores)	27	Power-good LED

13	Five pushbuttons	28	Xilinx Artix-7 FPGA
14	Pmod port(s)	29	DDR2 memory
15	Temperature sensor		

Artix-7

Artix-7 merupakan salah satu seri board FPGA (Field Programmable Gate Array) yang diproduksi oleh Xilinx. Artix-7 dirancang khusus untuk memberikan performa yang optimal dengan biaya yang terjangkau. Seri ini sangat populer digunakan dalam berbagai aplikasi karena memiliki fitur yang lengkap dan memiliki konsumsi daya yang rendah, serta mudah untuk digunakan.

Nexys A7

Nexys™A7, yang berbasis pada FPGA Artix-7, memiliki performa dan desain yang berfokus untuk digunakan oleh pelajar. Dengan FPGA berkapasitas besar, memori eksternal yang luas, dan berbagai *port* seperti USB, Ethernet, dan lainnya, Nexys A7 dapat digunakan untuk mendesain rangkaian digital dasar hingga proses yang kompleks.

Tabel B. 2 Spesifikasi Nexys A-7

FPGA	XC7A100T-1CSG324C
I/O Interfaces	<ul style="list-style-type: none"> • USB-UART • One 10/100 Ethernet • USB OTG 2.0 • USB-UART bridge • 12-bit VGA • 3-axis accelerometer • PWM audio output • Temperature sensor • PDM microphone • USB HID
Memory	<ul style="list-style-type: none"> • 128 Mbyte DDR2 • 128 Mbit Serial Flash • Micro SD card connector
Display	2 4-digit 7-Segment display
Switch and LED	<ul style="list-style-type: none"> • 16 Slide switches • 16 LEDs

	<ul style="list-style-type: none"> • 2 tri-color LEDs • 5 Push-buttons
Clock	100 MHz crystal oscillator
Expansion Ports	<ul style="list-style-type: none"> • Pmod XADC signals • 4 Pmod ports

B.6. HDL (Hardware Description Language)

Salah satu metode untuk perancangan rangkaian adalah menggunakan HDL (*Hardware Description Language*). Nantinya, tiap-tiap komponen serta jalur yang menghubungkannya akan dideskripsikan lewat tulisan atau kode tertentu. Tiap vendor FPGA memiliki aturan mengenai penggunaan kode dalam hal implementasi di dalam FPGA. Namun, sejak sekitar 10 tahun lalu, telah muncul kode baru yang dapat diimplementasikan ke dalam semua jenis FPGA buatan vendor manapun. Kode baru tersebut ada 2 yakni Verilog dan VHDL. Baik verilog maupun VHDL ternyata lebih terkenal karena mudah dipahami dan dimengerti. Selanjutnya, dua kode ini kemudian menjadi acuan utama dalam proses implementasi rancangan rangkaian ke dalam FPGA (apapun jenis vendornya). Hingga saat ini, metode perancangan menggunakan HDL (baik verilog maupun VHDL) lebih banyak digunakan daripada metode *schematic*.

B.6.1. Verilog

Verilog adalah bahasa HDL yang dikembangkan oleh IEEE (*Institute of Electrical and Electronic Engineering*) dan formatnya berbentuk teks untuk mendeskripsikan rangkaian dan sistem elektronik. Dalam desain elektronik, Verilog digunakan dalam verifikasi melalui simulasi, analisis waktu, analisis uji (penilaian kesalahan), dan sintesis logika.

B.6.2. VHDL

Pada praktikum ini, kita akan akan menggunakan VHDL untuk membuat desain program kedalam FPGA tersebut. VHDL (*VHSC Hardware Description Language*); VHSIC (*Very High Speed Integrated Circuit*) adalah sebuah bahasa pemrograman yang dikembangkan oleh IEEE (*Institute of Electrical and Electronic Engineering*). VHDL juga digunakan sebagai bahasa pemrograman untuk simulasi rangkaian dari komponen-komponen digital. Pada VHDL, konsep serta sintaks banyak diperlukan untuk mengerti bagaimana rancangan VHDL sebagai bagian dari pemrograman FPGA. Dalam kebanyakan kasus, keputusan memilih dan menggunakan kode VHDL daripada kode Verilog atau SystemC, sangat tergantung pada pilihan perancang itu sendiri dan lebih kepada ketersediaan *software* pendukung serta kebutuhan perusahaan. Untuk mempelajari VHDL, terdapat 5 komponen penting dalam menulis desain programnya, yaitu :

1. Entity
2. Architecture
3. Configuration
4. Package Declaration

5. Package Body

Perbedaan bahasa Verilog dan VHDL

Verilog dan VHDL adalah bahasa HDL yang digunakan untuk merancang sirkuit digital. Keduanya memiliki tujuan yang sama, namun memiliki pendekatan yang sedikit berbeda. Verilog cenderung lebih intuitif dan mirip dengan bahasa pemrograman tingkat tinggi, sehingga lebih disukai oleh para *engineer* yang memiliki latar belakang *software*. Bahasa ini sering digunakan untuk merancang sirkuit pada tingkat yang lebih rendah. Di sisi lain, VHDL memiliki struktur yang lebih formal dan matematis, membuatnya lebih cocok untuk verifikasi dan simulasi yang kompleks. Bahasa ini sering digunakan dalam proyek-proyek besar dan kompleks yang membutuhkan tingkat abstraksi yang lebih tinggi.

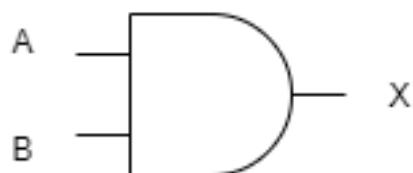
Tabel B. 3 Perbedaan Verilog dan VHDL

Verilog	VHDL
Bahasa HDL yang digunakan untuk memodelkan sistem elektronik.	Bahasa HDL yang digunakan untuk mendeskripsikan sistem digital dan campuran sinyal.
Berdasarkan bahasa C	Berdasarkan bahasa Pascal
Case Sensitive	Tidak Case Sensitive
Tidak mendukung Array multidimensi	Mendukung Array multidimensi
Verilog tidak mengizinkan pemanggilan tugas secara bersamaan	VHDL mengizinkan pemanggilan prosedur secara bersamaan.
Memiliki tipe data yang sederhana	Memiliki tipe data yang kompleks
Tidak memiliki Library	Terdapat Library

B.7. Jenis-jenis Gerbang Logika

a. Gerbang AND

Gerbang AND adalah gerbang logika dasar yang melakukan perkalian logika berdasarkan dari input yang diterapkan padanya. Gerbang ini menghasilkan output HIGH atau bernilai 1, hanya ketika semua input atau masukan bernilai 1. Jika tidak, output gerbang AND akan memiliki nilai LOW atau 0.



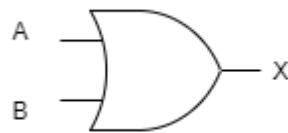
Gambar B. 11 Gerbang Logika AND

Tabel B. 4 Gerbang Logika AND

A	B	Out
0	0	0
0	1	0
1	0	0
1	1	1

b. Gerbang OR

Gerbang OR adalah gerbang logika dasar yang melakukan penjumlahan logika berdasarkan nilai dari input yang diterapkan pada gerbang. Gerbang ini akan menghasilkan nilai 1 apabila salah satu atau semua input memiliki nilai 1. Apabila tidak terdapat nilai 1 dari input, maka output gerbang OR akan bernilai 0.



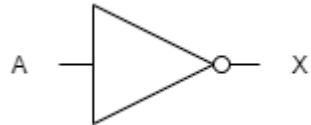
Gambar B. 12 Gerbang Logika OR

Tabel B. 5 Gerbang Logika OR

A	B	Out
0	0	0
0	1	1
1	0	1
1	1	1

c. Gerbang NOT

Gerbang NOT atau disebut juga sebagai inverter, akan menghasilkan output yang komplement dengan hasil inputnya. Jika inputnya bernilai 1 maka output akan bernilai 0 dan sebaliknya.



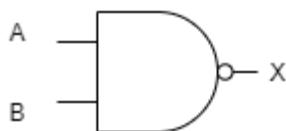
Gambar B. 13 Gerbang Logika NOT

Tabel B. 6 Gerbang Logika NOT

A	Out
1	0
0	1

d. Gerbang NAND

Gerbang NAND adalah gabungan dari gerbang logika AND dan juga gerbang logika NOT. Hasil dari gerbang logika NAND adalah kebalikan dari gerbang AND.



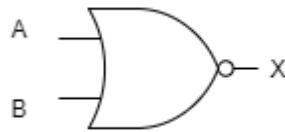
Gambar B. 14 Gerbang Logika NAND

Tabel B. 7 Gerbang Logika NAND

A	B	Out
0	0	1
0	1	1
1	0	1
1	1	0

e. Gerbang NOR

Gerbang NOR adalah gabungan dari gerbang logika OR dan juga gerbang logika NOT. Hasil dari gerbang logika NOR adalah kebalikan dari gerbang OR.



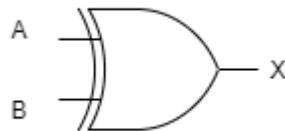
Gambar B. 15 Gerbang Logika NOR

Tabel B. 8 Gerbang Logika NOR

A	B	Out
0	0	1
0	1	0
1	0	0
1	1	0

f. Gerbang XOR

Gerbang XOR (*Exclusive OR*) adalah gerbang logika paritas ganjil yang akan menghasilkan nilai 1 apabila kedua inputnya memiliki nilai yang berbeda.



Gambar B. 16 Gerbang Logika XOR

Tabel B. 9 Gerbang Logika XOR

A	B	Out
0	0	0
0	1	1
1	0	1
1	1	0

g. Gerbang XNOR

Gerbang XNOR (*Exclusive NOR*) adalah gerbang logika paritas genap yang akan menghasilkan nilai 1 apabila kedua inputnya memiliki nilai yang sama.



Gambar B. 17 Gerbang Logika XNOR

Tabel B. 10 Gerbang Logika XNOR

A	B	Out
0	0	1
0	1	0
1	0	0
1	1	1