

# IMPLEMENTASI COMPARATOR 8-BIT MENGGUNAKAN VHDL PADA PERANGKAT LUNAK XILINK ISE

**Adkhan Sholeh**

Program Studi Manajemen Informatika  
STMIK Jenderal Achmad Yani Yogyakarta

[adkhan75@live.com](mailto:adkhan75@live.com)

## **Abstrak**

*Desain rangkaian elektronika dengan memanfaatkan perangkat lunak telah mengalami kemajuan yang pesat. VHDL, yang pada awalnya digunakan untuk mendeskripsikan karakteristik dan perilaku komponen elektronika (khususnya IC atau integrated circuit) berkembang menjadi alat untuk melakukan simulasi dan sintesa gerbang-gerbang logika dan komponen elektronik lainnya. Menggunakan VHDL, perancangan rangkaian elektronika dilakukan dengan mendeskripsikan entitas rangkaian dan arsitektur rangkaian.*

*Dalam berbagai bidang pengolahan data/sinyal, hampir bisa dipastikan adanya penggunaan comparator untuk membandingkan 2 (dua) atau lebih sinyal/data. Pada makalah ini disajikan bagaimana sebuah comparator 8 bit dirancang menggunakan VHDL dan simulasi hasil implementasinya pada platform FPGA (field programmable gate array).*

**Kata Kunci:** *comparator, VHDL, desain, rangkaian elektronika.*

## **1. Pendahuluan**

Proses perancangan menggunakan komputer (*computer-aided design*) telah dilakukan di berbagai bidang, termasuk di bidang elektronika digital yang telah berlangsung lebih dari satu dasa warsa. Di tahun 1970-an hingga 1980-an, perancangan rangkaian banyak dilakukan dengan pendekatan skematik. Pendekatan ini mengalami pergeseran pada pertengahan tahun 1990-an, di mana perancangan rangkaian mulai dilakukan menggunakan perangkat lunak berbasis *hardware design language* (HDL).

Perancangan berbasis perangkat lunak pada awalnya dimaksudkan untuk menggantikan penggunaan komponen elektronik yang cenderung mahal dan menghindari resiko kerusakan. Kini, penggunaan perangkat lunak ini juga dipandang sebagai solusi atas kerumitan desain rangkaian elektronika digital. Perancangan rangkaian digital yang melibatkan ribuan hingga jutaan komponen diskret (yaitu gerbang-gerbang logika digital) merupakan hal yang sangat rumit untuk dikerjakan. Perancangan dan uji cobanya secara konvensional tentu memerlukan biaya yang sangat besar dengan resiko kerusakan komponen yang nyata.

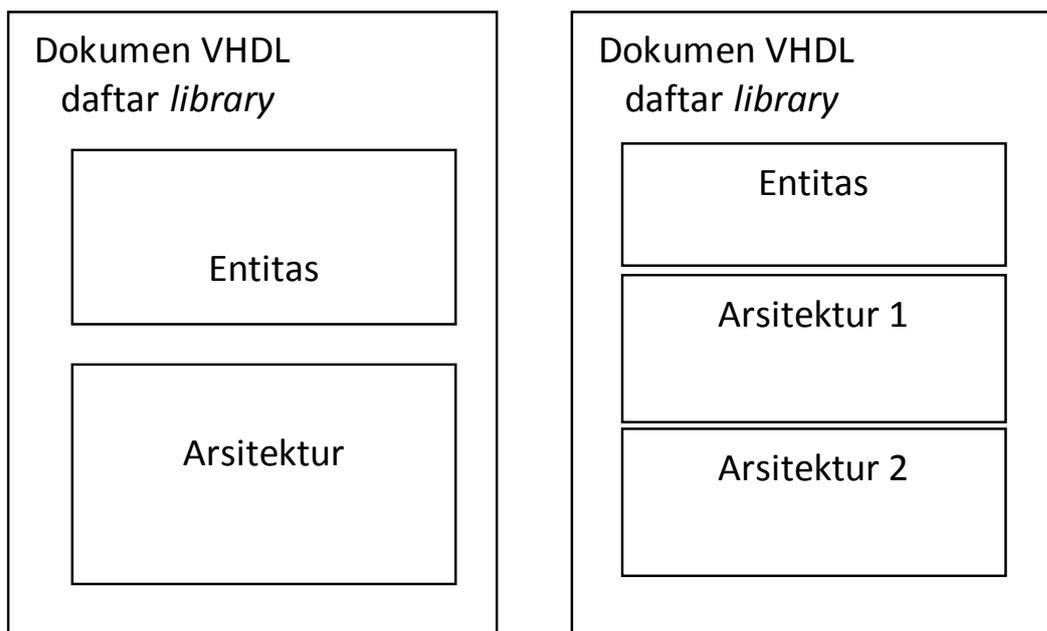
## 2. Dasar Teori

### 2.1 VHDL

VHDL merupakan singkatan dari VHSIC HDL (Very High Speed Integrated Circuit HDL). VHDL merupakan bahasa yang berkembang dari fungsi awalnya sebagai alat untuk mendeskripsikan karakteristik dan perilaku *chip* IC, menjadi bahasa untuk perancangan rangkaian dan sistem elektronika. VHDL kini menjadi bahasa pemrograman standar untuk otomasi perancangan rangkaian elektronika. Pada tingkat perancangan, VHDL mempunyai kemampuan untuk mendeskripsikan sistem yang menggunakan sinyal digital maupun yang menggunakan sinyal campuran.

Awalnya, VHDL dikembangkan atas perintah Departemen Pertahanan Amerika Serikat untuk kepentingan mendokumentasikan perilaku *chip* ASIC (Application Specific Integrated Circuit). Lahirnya dokumen-dokumen ini memunculkan keinginan agar dokumentasi tersebut dapat dibaca oleh simulator gerbang logika yang saat itu telah ada. Langkah tersebut kemudian dilanjutkan dengan pengembangan alat-alat sintesa gerbang logika yang mampu membaca dokumen VHDL, dan menghasilkan output definisi implementasi fisik rangkaian.

Sebuah dokumen VHDL selalu terdiri atas dua bagian, yaitu entitas dan arsitektur. Sebuah deskripsi entitas dapat berpasangan dengan sebuah deskripsi arsitektur. Adapun arsitektur, sebuah dokumen berisi rancangan VHDL bisa menampung lebih dari satu arsitektur.



**Gambar 1:** Struktur VHDL  
(a) dengan 1 deskripsi arsitektur (b) dengan 2 deskripsi arsitektur

Arsitektur sebuah rangkaian dapat dijabarkan melalui dua cara, yaitu:

- menggunakan skematik
- menggunakan HDL

## 2.2 Comparator

*Comparator* adalah rangkaian/komponen elektronika yang berfungsi untuk membandingkan sinyal-sinyal yang diterimanya melalui *input*. Pada jenis komparator digital, hasil perbandingan 2 data digital (misalnya disebut *A* dan *B*) akan menghasilkan satu dari 3 (tiga) keluaran:

- $A = B$  (nilai *A* sama dengan nilai *B*)
- $A < B$  (nilai *A* lebih kecil dari nilai *B*)
- $A > B$  (nilai *A* lebih besar dari nilai *B*)

Berikut ini disajikan rancangan *comparator* 8 bit dalam bahasa VHDL. Baris pertama dan kedua dokumen menyatakan *library* (pustaka) yang digunakan dalam rancangan. Bagian tersebut diikuti dengan deskripsi entitas yang menyatakan nama rangkaian sebagai *comp*. Entitas *comp* memuat 2 (dua) *input* yaitu *x* dan *y*, masing-masing sebagai vektor (*multiple line* atau *bus*) yang lebarnya masing-masing adalah 8 bit (dari 7 sampai 0). Input *x* dan *y* inilah yang akan dibandingkan oleh rangkaian *comparator* dimaksud. Entitas ini juga memuat spesifikasi 3 jalur *output*, yaitu *gt*, *eq*, dan *lt*. Output *gt* untuk mengindikasikan hasil *greater than* ( $x > y$ ), *eq* untuk mengindikasikan *equal* ( $x = y$ ), dan *lt* mengindikasikan *less than* ( $x < y$ ).

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
entity comp is
  port(
    x : in STD_LOGIC_VECTOR(7 downto 0);
    y : in STD_LOGIC_VECTOR(7 downto 0);
    gt : out STD_LOGIC;
    eq : out STD_LOGIC;
    lt : out STD_LOGIC
  );
end comp;

architecture comp of comp is
begin
  process(x, y)
  begin
    gt <= '0';
    eq <= '0';
    lt <= '0';
    if (x > y) then
      gt <= '1';
    elsif (x = y) then
```

```

    eq <= '1';
    elsif (x < y) then
        lt <= '1';
    end if;
end process;
end comp;

```

Bagian selanjutnya yaitu arsitektur, yang menggunakan nama *comp*. Arsitektur ini mendeskripsikan aktivitas pemrosesan input  $x$  dan  $y$  dengan deskripsi perintah `process(x, y)`. Pada rancangan ini, process dimulai dengan melakukan inisialisasi pada ketiga jalur *output*. Instruksi berikutnya adalah melakukan pengecekan dengan perintah logika `if` atas 3 (tiga) kemungkinan kombinasi *input*, kemudian mengeluarkan *output* yang sesuai. Bagian ini ditutup berturut-turut dengan `end if`, `end process`, dan penutup deskripsi arsitektur `end comp`.

### 3. Analisa

Rancangan di atas diimplementasikan menggunakan perangkat lunak Xilinx ISE beserta simulator Xilinx ISim. Implementasi dilakukan dengan tahapan:

- Sintesis rancangan pada Xilinx ISE

Pada tahap sintesis dapat diketahui adanya kesalahan-kesalahan rancangan yang bersumber pada penggunaan *library* yang tidak lengkap/tepat, deskripsi entitas maupun arsitektur yang bermasalah. Dengan *platform* target *chip* FPGA xc3s700a-5fg400, berikut ini adalah hasil sintesa rangkaian *comparator*, yang menunjukkan hasil implementasi *Placed and Routed*.

comp Project Status (10/25/2012 - 13:44:36)			
<b>Project File:</b>	tugas_comparator.xise	<b>Parser Errors:</b>	No Errors
<b>Module Name:</b>	comp	<b>Implementation State:</b>	Placed and Routed
<b>Target Device:</b>	xc3s700a-5fg400	<b>• Errors:</b>	
<b>Product Version:</b>	ISE 13.2	<b>• Warnings:</b>	
<b>Design Goal:</b>	Balanced	<b>• Routing Results:</b>	<a href="#">All Signals Completely Routed</a>
<b>Design Strategy:</b>	<a href="#">Xilinx Default (unlocked)</a>	<b>• Timing Constraints:</b>	
<b>Environment:</b>	<a href="#">System Settings</a>	<b>• Final Timing Score:</b>	0 ( <a href="#">Timing Report</a> )

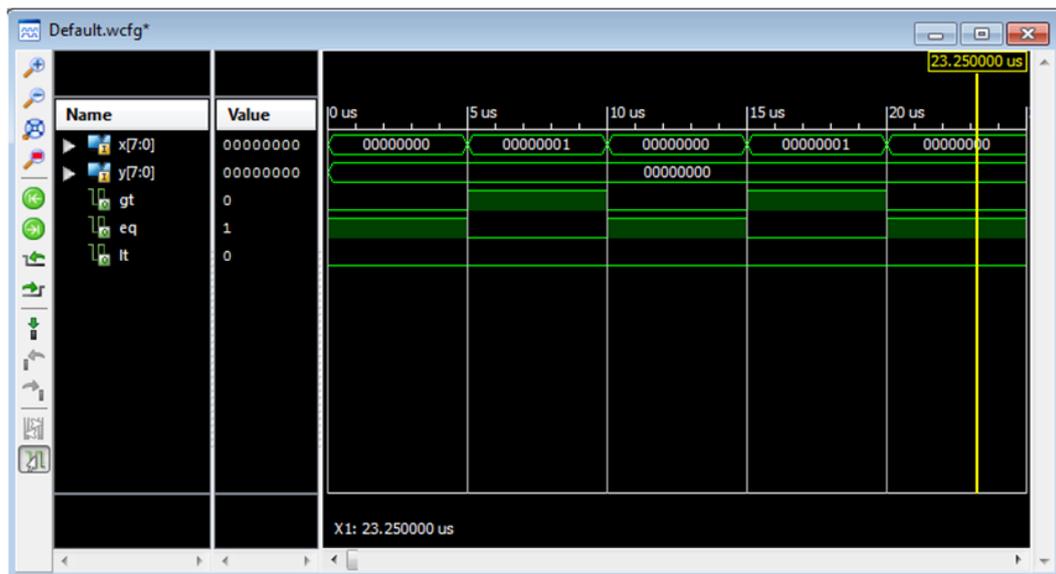
**Gambar 2:** Status implementasi *project comparator*

Selain status implementasi di atas, perangkat lunak juga menyajikan laporan utilisasi atas *chip* FPGA yang digunakan sebagai *platform* rancangan. Secara umum, rancangan *comparator* yang dibuat hanya menggunakan sedikit dari sumber daya *chip* FPGA yang tersedia.

Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Number of 4 input LUTs	23	11,776	1%	
Number of occupied Slices	14	5,888	1%	
Number of Slices containing only related logic	14	14	100%	
Number of Slices containing unrelated logic	0	14	0%	
Total Number of 4 input LUTs	23	11,776	1%	
Number of bonded IOBs	19	311	6%	
Average Fanout of Non-Clock Nets	2.13			

**Gambar 3:** Ringkasan utilisasi sumber daya *chip*

- Simulasi rancangan pada Xilinx ISim  
Setelah memastikan bahwa implementasi pada tahap sintesa tidak mengandung *error* dan *warning*, maka proses perancangan dilanjutkan dengan mensimulasikan desain *comparator*. Beberapa kondisi sinyal yang ditampilkan antara lain:

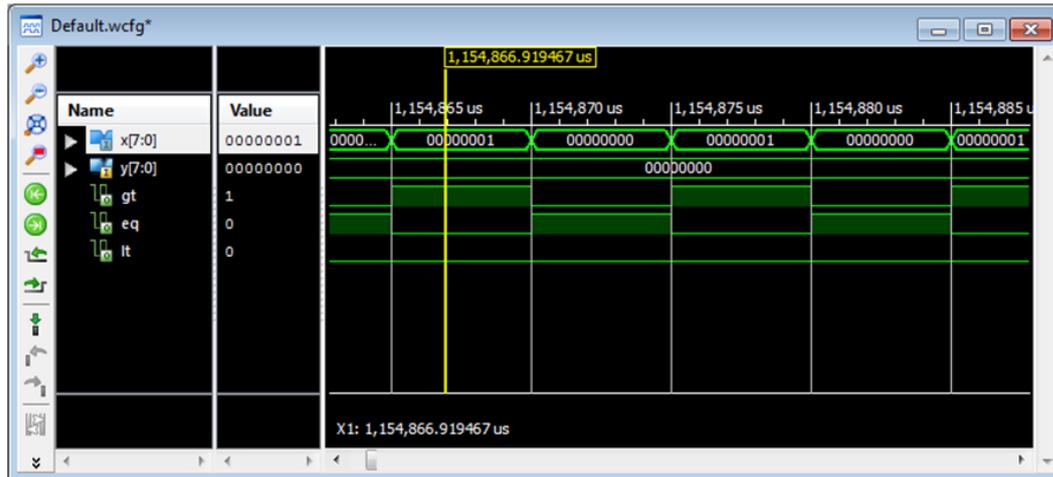


**Gambar 4:** Kombinasi *input*  $x = y$  menyebabkan  $eq = 1$

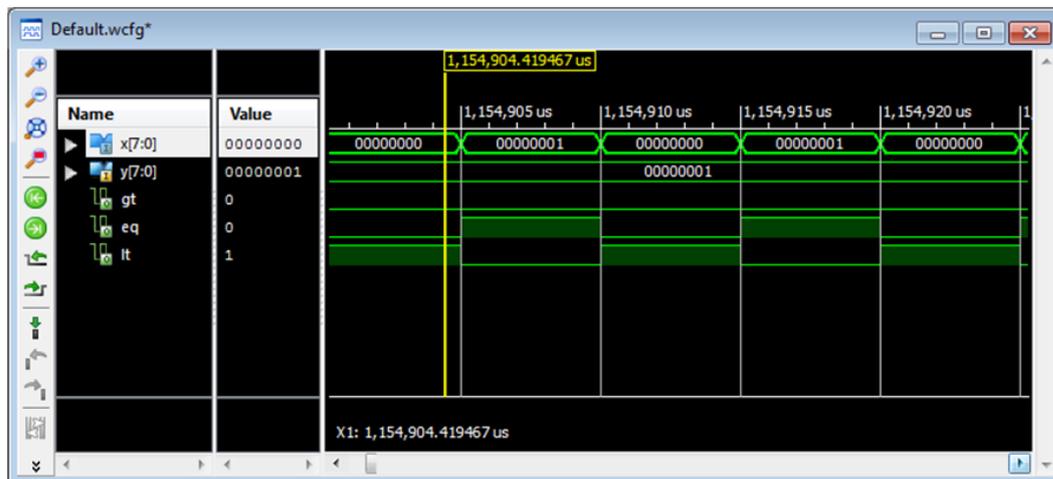
Gambar 4 menunjukkan keadaan ketika  $x$  dan  $y$  bernilai '00000000', maka diperoleh output  $eq$  (*equal*) = '1' dan lainnya '0'.

Gambar 5 menunjukkan keadaan ketika  $x$  bernilai '00000001' dan  $y$  bernilai '00000000', maka diperoleh output  $gt$  (*greater than*) = '1' dan lainnya '0'.

Gambar 6 menunjukkan keadaan ketika  $x$  bernilai '00000000' dan  $y$  bernilai '00000001', maka diperoleh output  $lt$  (*less than*) = '1' dan lainnya '0'.



Gambar 5: Kombinasi *input*  $x > y$  menyebabkan  $gt = 1$



Gambar 6: Kombinasi *input*  $x < y$  menyebabkan  $lt = 1$

#### 4. Kesimpulan

1. Sebuah rangkaian elektronika dengan fungsi sebagai *comparator* dapat dibangun menggunakan VHDL.
2. Rancangan *comparator* tersebut telah disintesa dan disimulasikan pada perangkat lunak Xilinx, dapat menerima *input* serta menghasilkan *output* dengan perilaku yang sesuai prediksi.

#### Daftar Pustaka

- Dubey, R., 2009, *Introduction to Embedded System Design Using FPGA*, Springer, London.
- Haskell, R.E. dan Hanna, D.M., 2009, *Introduction to Digital Design Using Digilent FPGA Boards*, LBE Books, Michigan.
- Putro, A.E., 2011, *Embedded System*, Modul Kuliah, FMIPA UGM, Yogyakarta.