

PETUNJUK PRAKTIKUM

Praktikum

Sistem Digital



**Laboratorium Dasar
Teknik Elektro**

**Sekolah Teknik Elektro Dan Informatika
Institut Teknologi Bandung
2024**

Buku Petunjuk

Praktikum Sistem Digital

EL 2102

Percobaan II

Rangkaian Logika Kombinasional

v.1.1

Mervin T. Hutabarat

Arif Sasongko

Eric Agustian

Harry Septanto

M. Zakiyullah R.

Ardimas Andi Purwita

Nina Lestari

Sekolah Teknik Elektro Dan Informatika

Institut Teknologi Bandung

2024

DAFTAR ISI

RIWAYAT DOKUMEN.....	IV
ATURAN LABORATORIUM.....	V
ATURAN UMUM LABORATORIUM	V
KELENGKAPAN	V
PERSIAPAN	V
PERGANTIAN JADWAL	VI
SANKSI	VI
PANDUAN UMUM KESELAMATAN DAN PENGGUNAAN PERALATAN LABORATORIUM	VII
KESELAMATAN	VII
PENGGUNAAN PERALATAN PRAKTIKUM	VIII
SANKSI	VIII
TABEL SANKSI PRAKTIKUM.....	IX
PERCOBAAN II – RANGKAIAN LOGIKA KOMBINASIONAL.....	11
1.1 TUJUAN.....	11
CATATAN	11
1.2 PERSIAPAN	11
1.3 DASAR TEORI	11
IMPLEMENTASI FPGA DAN WAKTU TUNDA	11
FULL ADDER	12
BCD-TO-7-SEGMENT CODE CONVERTER	13
5BIT BINARY-TO-BCD CODE CONVERTER	14
4BIT ADDER WITH 7 SEGMENT	14
1.4 TUGAS PENDAHULUAN	15
1.5 PERCOBAAN	16
PERALATAN YANG DIGUNAKAN.....	16
PERCOBAAN 2A: IMPLEMENTASI BCD-TO-7SEGMENT PADA FPGA BOARD	16
PERCOBAAN 2B: MERANCANG BCD-TO-7SEGMENT DENGAN LEVEL ABSTRAKSI BEHAVIORAL... 16	16
PERCOBAAN 2C: MEMBUAT RANGKAIAN 4BIT ADDER WITH 7 SEGMENT	18
1.6 MENGAKHIRI PERCOBAAN	20
APENDIKS.....	21
DE10-LITE FPGA BOARD	21
PIN ASSIGNMENT OF SLIDE SWITCHES.....	22
PIN ASSIGNMENT OF LEDS.....	23
PIN ASSIGNMENT OF 7-SEGMENT DISPLAYS	24
CONTOH PEMBERIAN COMMENT PADA VHDL	26

TABEL PENGUJIAN BCD-TO-7SEGMENT (PERCOBAAN 2A DAN 2B)	27
TABEL PENGUJIAN 4BIT ADDER WITH 7 SEGMENT.....	28

RIWAYAT DOKUMEN

VERSI	TANGGAL	CATATAN PERUBAHAN
1.1	8 Oktober 2024	Menambahkan gambar FPGA board, tabel pin assignment di apendiks. Menambahkan tabel pengujian di Apendiks.
1.0	7 Oktober 2024	Pembuatan dokumen.

ATURAN LABORATORIUM

ATURAN UMUM LABORATORIUM

KELENGKAPAN

Setiap praktikan wajib berpakaian lengkap, mengenakan **celana panjang/ rok, kemeja** dan mengenakan **sepatu**. Untuk memasuki ruang laboratorium Praktikan wajib membawa kelengkapan berikut:

- Modul praktikum
- Buku Catatan Laboratorium (BCL)
- Alat tulis (dan kalkulator, jika diperlukan)
- *Name tag*
- Kartu Praktikum

PERSIAPAN

SEBELUM PRAKTIKUM

Sebelum mengikuti percobaan sesuai jadwalnya, sebelum memasuki laboratorium praktikan harus mempersiapkan diri dengan melakukan hal-hal berikut:

- Membaca dan memahami isi modul praktikum,
- Mengerjakan Tugas Pendahuluan
- Mengerjakan hal-hal yang harus dikerjakan sebelum praktikum dilaksanakan, misalnya mengerjakan perhitungan-perhitungan, menyalin source code, mengisi Kartu Praktikum dsb.,
- Mengisi daftar hadir di Tata Usaha Laboratorium,
- Mengambil kunci loker dan melengkapi administrasi peminjaman kunci loker dengan meninggalkan kartu identitas (KTM/ SIM/ KTP).

SELAMA PRAKTIKUM

Setelah dipersilahkan masuk dan menempati bangku dan meja kerja, praktikan haruslah:

- Memperhatikan dan mengerjakan setiap percobaan dengan waktu sebaik-baiknya, diawali dengan kehadiran praktikan secara tepat waktu,
- Mengumpulkan Kartu Praktikum pada asisten,
- Mendokumentasikan dalam Buku Catatan Laboratorium. (lihat Petunjuk Penggunaan BCL) tentang hal-hal penting terkait percobaan yang sedang dilakukan.

SETELAH PRAKTIKUM

- Memastikan BCL telah ditandatangani oleh asisten,
- Mengembalikan kunci loker dan melengkapi administrasi pengembalian kunci loker (pastikan kartu identitas KTM/ SIM/ KTP diperoleh kembali),
- Mengerjakan laporan dalam bentuk SoftCopy (lihat Panduan Penyusunan Laporan di laman <https://ldte.stei.itb.ac.id/panduan/>)),

- Mengirimkan file laporan dengan cara mengunggah di laman <https://praktikum.stei.itb.ac.id>. Waktu pengiriman paling lambat jam 11.00 WIB, dua hari kerja berikutnya setelah praktikum, kecuali ada kesepakatan lain antara Dosen Pengajar dan/ atau Asisten.

PERGANTIAN JADWAL

KASUS BIASA

Pertukaran jadwal hanya dapat dilakukan per orang dengan modul yang sama. Langkah untuk menukar jadwal adalah sebagai berikut:

Lihatlah format Pertukaran Jadwal di <https://ldte.stei.itb.ac.id/panduan/> pada halaman Panduan

- Salah satu praktikan yang bertukar jadwal harus mengirimkan e-mail ke labdasar@stei.itb.ac.id . Waktu pengiriman paling lambat jam 16.30, sehari sebelum praktikum yang dipertukarkan
- Pertukaran diperbolehkan setelah ada email konfirmasi dari Lab. Dasar

KASUS SAKIT ATAU URUSAN MENDESAK PRIBADI LAINNYA

Jadwal pengganti dapat diberikan kepada praktikan yang sakit atau memiliki urusan mendesak pribadi.

- Praktikan yang hendak mengubah jadwal untuk urusan pribadi mendesak harus memberitahu staf tata usaha laboratorium sebelum jadwal praktikumnya melalui email.
- Segera setelah praktikan memungkinkan mengikuti kegiatan akademik, praktikan dapat mengikuti praktikum pengganti setelah mendapatkan konfirmasi dari staf tata usaha laboratorium dengan melampirkan surat keterangan dokter bagi yang sakit atau surat terkait untuk yang memiliki urusan pribadi.

KASUS "KEPENTINGAN MASSAL"

"Kepentingan massal" terjadi jika ada lebih dari 1/3 rombongan praktikan yang tidak dapat melaksanakan praktikum pada satu hari yang sama karena alasan yang terkait kegiatan akademis.

SANKSI

Pengabaian aturan-aturan di atas dapat dikenakan sanksi pengguguran nilai praktikum terkait.

PANDUAN UMUM KESELAMATAN DAN PENGGUNAAN PERALATAN LABORATORIUM

KESELAMATAN

Pada prinsipnya, untuk mewujudkan praktikum yang aman diperlukan partisipasi seluruh praktikan dan asisten pada praktikum yang bersangkutan. Dengan demikian, kepatuhan setiap praktikan terhadap uraian panduan pada bagian ini akan sangat membantu mewujudkan praktikum yang aman.

BAHAYA LISTRIK

- Perhatikan dan pelajari tempat-tempat sumber listrik (stop-kontak dan *circuit breaker*) dan cara menyala-matikannya. Jika melihat ada kerusakan yang berpotensi menimbulkan bahaya, laporan pada asisten
- Hindari daerah atau benda yang berpotensi menimbulkan bahaya listrik (sengatan listrik/ strum) secara tidak disengaja, misalnya kabel jala-jala yang terkelupas dll.
- Tidak melakukan sesuatu yang dapat menimbulkan bahaya listrik pada diri sendiri atau orang lain
- Keringkan bagian tubuh yang basah karena, misalnya, keringat atau sisa air wudhu
- Selalu waspada terhadap bahaya listrik pada setiap aktivitas praktikum

Kecelakaan akibat bahaya listrik yang sering terjadi adalah tersengat arus listrik. Berikut ini adalah hal-hal yang harus diikuti praktikan jika hal itu terjadi:

- Jangan panik
- Matikan semua peralatan elektronik dan sumber listrik di meja masing-masing dan di meja praktikan yang tersengat arus listrik
- Bantu praktikan yang tersengat arus listrik untuk melepaskan diri dari sumber listrik
- Beritahukan dan minta bantuan asisten, praktikan lain dan orang di sekitar anda tentang terjadinya kecelakaan akibat bahaya listrik

BAHAYA API ATAU PANAS BERLEBIH

- Jangan membawa benda-benda mudah terbakar (korek api, gas dll.) ke dalam ruang praktikum bila tidak disyaratkan dalam modul praktikum
- Jangan melakukan sesuatu yang dapat menimbulkan api, percikan api atau panas yang berlebihan
- Jangan melakukan sesuatu yang dapat menimbulkan bahaya api atau panas berlebih pada diri sendiri atau orang lain
- Selalu waspada terhadap bahaya api atau panas berlebih pada setiap aktivitas praktikum

Berikut ini adalah hal-hal yang harus diikuti praktikan jika menghadapi bahaya api atau panas berlebih:

- Jangan panik
- Beritahukan dan minta bantuan asisten, praktikan lain dan orang di sekitar anda tentang terjadinya bahaya api atau panas berlebih
- Matikan semua peralatan elektronik dan sumber listrik di meja masing-masing
- Menjauh dari ruang praktikum

BAHAYA BENDA TAJAM DAN LOGAM

- Dilarang membawa benda tajam (pisau, gunting dan sejenisnya) ke ruang praktikum bila tidak diperlukan untuk pelaksanaan percobaan
- Dilarang memakai perhiasan dari logam misalnya cincin, kalung, gelang dll.
- Hindari daerah, benda atau logam yang memiliki bagian tajam dan dapat melukai
- Tidak melakukan sesuatu yang dapat menimbulkan luka pada diri sendiri atau orang lain

LAIN-LAIN

- Dilarang membawa makanan dan minuman ke dalam ruang praktikum

PENGGUNAAN PERALATAN PRAKTIKUM

Berikut ini adalah panduan yang harus dipatuhi ketika menggunakan alat-alat praktikum:

- Sebelum menggunakan alat-alat praktikum, pahami petunjuk penggunaan alat itu. Petunjuk penggunaan beberapa alat dapat didownload di <http://labdasar.ee.itb.ac.id>
- Perhatikan dan patuhi peringatan (*warning*) yang biasa tertera pada badan alat
- Pahami fungsi atau peruntukan alat-alat praktikum dan gunakanlah alat-alat tersebut hanya untuk aktivitas yang sesuai fungsi atau peruntukannya. Menggunakan alat praktikum di luar fungsi atau peruntukannya dapat menimbulkan kerusakan pada alat tersebut dan bahaya keselamatan praktikan
- Pahami *rating* dan jangkauan kerja alat-alat praktikum dan gunakanlah alat-alat tersebut sesuai *rating* dan jangkauan kerjanya. Menggunakan alat praktikum di luar *rating* dan jangkauan kerjanya dapat menimbulkan kerusakan pada alat tersebut dan bahaya keselamatan praktikan
- Pastikan seluruh peralatan praktikum yang digunakan aman dari benda/ logam tajam, api/ panas berlebih atau lainnya yang dapat mengakibatkan kerusakan pada alat tersebut
- Tidak melakukan aktifitas yang dapat menyebabkan kotor, coretan, goresan atau sejenisnya pada badan alat-alat praktikum yang digunakan

SANKSI

Pengabaian uraian panduan di atas dapat dikenakan sanksi tidak lulus mata kuliah praktikum yang bersangkutan

TABEL SANKSI PRAKTIKUM

Berlaku mulai: 14 Agustus 2017

Level	Waktu	Kasus	Sanksi	Pengurangan nilai per modul
Akademik	Saat dan setelah praktikum	Semua kegiatan plagiasi (mencontek): tugas pendahuluan, test dalam praktikum, laporan praktikum	Gugur praktikum	
		Sengaja tidak mengikuti praktikum		
Berat	Saat praktikum	Tidak hadir praktikum	Gugur modul	
		Terlambat hadir praktikum		
		Pakaian tidak sesuai: kemeja, sepatu		
		Tugas pendahuluan tidak dikerjakan/hilang/tertinggal		
Ringan	Saat Praktikum	Pertukaran jadwal tidak sesuai aturan/ketentuan		-25 nilai akhir
		Tidak mempelajari modul sebelum praktikum/tidak mengerti isi modul	Dikeluarkan dari praktikum	-25 nilai akhir
		BCL tertinggal/hilang		-100% nilai BCL
		Name Tag tertinggal/hilang		-10 nilai akhir
		Kartu praktikum tertinggal/hilang		-25 nilai akhir
		Kartu praktikum tidak lengkap data dan foto		-10 nilai akhir
		Loker tidak dikunci/kunci tertinggal		-10 nilai akhir

Setelah Praktikum	Tidak ada paraf asisten di BCL/kartu praktikum		-25 nilai akhir
	Terlambat mengumpulkan laporan		-1/min nilai akhir, maks -50
	Terlambat mengumpulkan BCL		-1/min nilai BCL, maks -50
	Tidak bawa kartu praktikum saat pengumpulan BCL		-50 nilai BCL
	Tidak minta paraf admin saat pengumpulan BCL		-50 nilai BCL

Catatan:

1. Pelanggaran akademik menyebabkan gugur praktikum, nilai praktikum E
2. Dalam satu praktikum, praktikan maksimal boleh melakukan
 - a. 1 pelanggaran berat dan 1 pelanggaran ringan; atau
 - b. 3 pelanggaran ringan
3. Jika jumlah pelanggaran melewati point 2, praktikan dianggap gugur praktikum.
4. Praktikan yang terkena sanksi gugur modul wajib mengganti praktikum pada hari lain dengan nilai modul tetap 0. Waktu pengganti praktikum ditetapkan bersama asisten. Jika praktikan tidak mengikuti ketentuan praktikum (pengganti) dengan baik, akan dikenakan sanksi gugur praktikum.
5. Setiap pelanggaran berat dan ringan dicatat/diberikan tanda di kartu praktikum
6. Waktu acuan adalah waktu sinkron dengan NIST
7. Sanksi yang tercantum di tabel adalah sanksi minimum.
8. Sanksi yang belum tercantum akan ditentukan kemudian.

PERCOBAAN II – RANGKAIAN LOGIKA KOMBINASIONAL

1.1 TUJUAN

1. Mendesain rangkaian kombinasional 4bit Adder dengan 7 Segment menggunakan blok-blok fungsi pembangun.
2. Melakukan implementasi rangkaian kombinasional pada FPGA board.
3. Mengenal level abstraksi dalam perancangan digital.

CATATAN

Untuk seluruh percobaan 2 ini, jika Anda menggunakan design skematik sesuai yang tertera di modul, maka nilai maksimal yang bisa Anda dapatkan adalah 75. Jika Anda menggunakan VHDL, maka nilai maksimal yang Anda dapatkan adalah 100. Code VHDL telah dipersiapkan praktikan sebelum praktikum. Jika ada pelanggaran berupa copy paste sebagian atau seluruh code VHDL, praktikan dikenai sanksi nilai nol dan/atau tidak lulus praktikum. Nilai tambah akan diberikan jika praktikan menggunakan simulator Modelsim®.

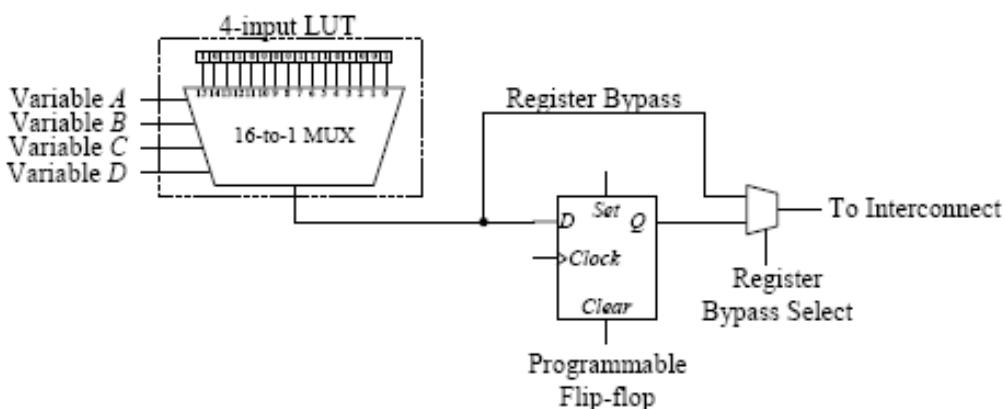
1.2 PERSIAPAN

Pelajari kembali bahan kuliah Anda mengenai rangkaian logika kombinasional. Pelajari juga keseluruhan petunjuk praktikum untuk modul rangkaian logika kombinasional ini. Kerjakan **Tugas Pendahuluan** dan kumpulkan sesuai ketentuan yang berlaku.

1.3 DASAR TEORI

IMPLEMENTASI FPGA DAN WAKTU TUNDA

Dalam teknologi FPGA MAX10 yang kita gunakan, fungsi logika diuraikan oleh software implementasi ke dalam bentuk subfungsi 4-masukan. Setiap subfungsi kemudian diimplementasikan oleh tabel kebenaran yang bekerja seperti multiplexer dan dibuat dengan memprogram SRAM yang mendefinisikan fungsionalitas dari FPGA. Setiap tabel kebenaran memiliki waktu tunda yang berkontribusi ke waktu tunda keseluruhan. Sedangkan untuk membedakan antara rangkaian kombinasional dan sekuensial, dalam subfungsi juga diberikan sebuah D flip-flop seperti yang terlihat pada Gambar 1.



Gambar 1. Bentuk subfungsi yang merepresentasikan logika pada FPGA

Penguraian kedalam subfungsi yang dikombinasikan dengan routing interkoneksi menghasilkan ketidakpastian dalam delay propagasi dari masukan ke keluaran dalam implementasi rangkaian. Suatu persamaan logika dengan

2 variabel mungkin saja memiliki waktu tunda yang sama dengan yang menggunakan 4 variabel karena bentuk subfungsi FPGA.

Perancang yang berpengalaman mungkin bisa menggunakan pengaturan tertentu untuk menspesifikasi waktu tunda maksimum yang dapat diterima. Apapun masalahnya, sangat berguna bagi kita untuk mengetahui berapa waktu tunda dari rangkaian kita. Karena hampir semua rangkaian kombinasional ditempatkan pada kondisi sekuensial, biasanya kita tertarik pada *worst case delay* yang bisa terjadi dalam operasi rangkaian dari masukan rangkaian kombinasional ke setiap keluaran rangkaian kombinasional.

Estimasi *worst case delay* ditentukan dengan menambahkan *delay* perkiraan maksimum kedalam rangkaian kombinasional termasuk logika dan interkoneksi. Karena ketidakpastian ini, *worst case delay* hanya bisa ditentukan setelah proses implementasi selesai termasuk penguraian menjadi subfungsi dan routing interkoneksi.

Dalam percobaan ini, kita akan membangun dua rangkaian. Dengan rangkaian pertama kita akan melihat beberapa tipe dari simulasi yang dapat kita gunakan dan melihat kemungkinan efek dari proses penguraian yang mengimplementasikan rangkaian sebenarnya secara fisik. Kemudian dengan rangkaian kedua, selain memverifikasi fungsionalitasnya, kita juga akan mencari *worst case delay* dari setiap masukan ke setiap keluaran dan akan menggunakan metode simulasi yang hanya dapat diaplikasikan pada rangkaian sederhana untuk mencari jalur sebenarnya yang ditempuh dimana delay ditemukan. Dengan Mengetahui jalur dari *worst case delay* kita kemudian bisa mengukur delay pada setiap titik jalur tersebut di lab. Delay yang terukur ini bukanlah *worst case delay* tetapi lebih kepada waktu tunda rata-rata.

FULL ADDER

Keunggulan FULL-ADDER bila dibandingkan dengan HALF-ADDER adalah kemampuan-nya menampung dan menjumlahkan bit CARRY-in (Cin) yang berasal dari CARRY-out (Cout) dari tahapan sebelumnya. Oleh karenanya fungsi FULL ADDER itu sendiri adalah menjumlahkan ke-tiga bit input yaitu bit A, bit B dan Cin untuk menghasilkan dua bit output yaitu S dan Cout.

Dengan menginterpretasikan fungsi dan melihat format operasi rangkaian FULLADDER, tabel kebenaran dapat disusun untuk setiap kemungkinan kombinasi ketiga bit input. Diasumsikan input berasal dari sumber logika positif dan output berupa ACTIVE HIGH.

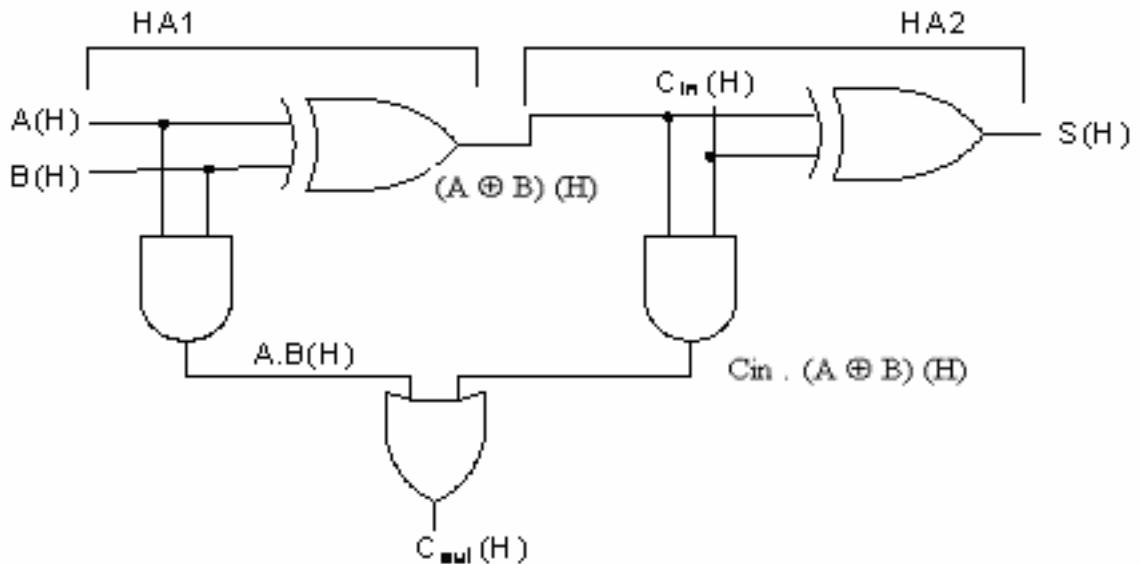
Langkah selanjutnya adalah membuat K-Map orde 2 dari tabel kebenaran tersebut. KMap ini akan membantu merumuskan fungsi logika dari S dan Cout.

Tabel 1. Tabel kebenaran dan K-map dari FULL ADDER

$S = C_{in}(A \oplus B) + \overline{C_{in}}(A \oplus B)$ $= A \oplus B \oplus C_{in}$ $C_{out} = C_{in}(A \oplus B) + (A \cdot B)$	<table border="1" style="margin-bottom: 10px;"> <thead> <tr> <th>A</th> <th>B</th> <th>C_{in}</th> <th>S</th> <th>C_{out}</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> </tbody> </table> <p>(a)</p>	A	B	C _{in}	S	C _{out}	0	0	0	0	0	0	0	1	1	0	0	1	0	1	0	0	1	1	0	1	1	0	0	1	0	1	0	1	0	1	1	1	0	0	1	1	1	1	1	1	<p>(b)</p>
A	B	C _{in}	S	C _{out}																																											
0	0	0	0	0																																											
0	0	1	1	0																																											
0	1	0	1	0																																											
0	1	1	0	1																																											
1	0	0	1	0																																											
1	0	1	0	1																																											
1	1	0	0	1																																											
1	1	1	1	1																																											

Implementasikan rangkaian FULL-ADDER dibuat berdasarkan persamaan ekspresi logika di atas. Rangkaian ini dapat tersusun dari dua buah HALF-ADDER (HA1 dan HA2), seperti terlihat pada **Gambar 2**.

Untuk penjumlahan dengan jumlah bit yang lebih banyak, dapat dilakukan dengan menambahkan rangkaian HALF ADDER, sesuai dengan jumlah bit input. Di pasaran, rangkaian FULL ADDER sudah ada yang berbentuk IC, seperti 74xx83 (4-bit FULL ADDER).



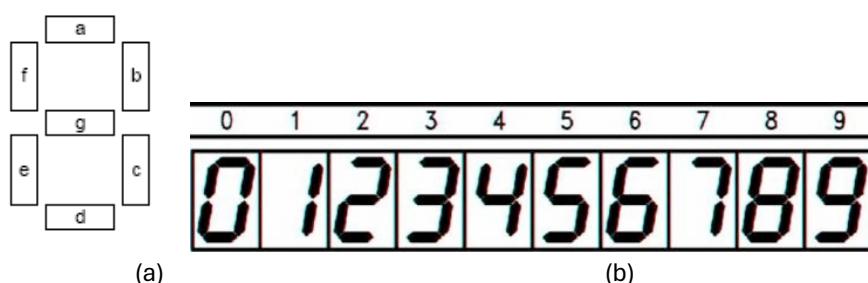
Gambar 2. Salah satu bentuk rangkaian Full Adder

Terdapat beberapa jenis rangkaian FULL ADDER, yaitu PARALLEL ADDER, LOOK AHEAD CARRY ADDER, dan CARRY SAVE ADDER dimana masing-masing memiliki kelebihan dan kekurangannya.

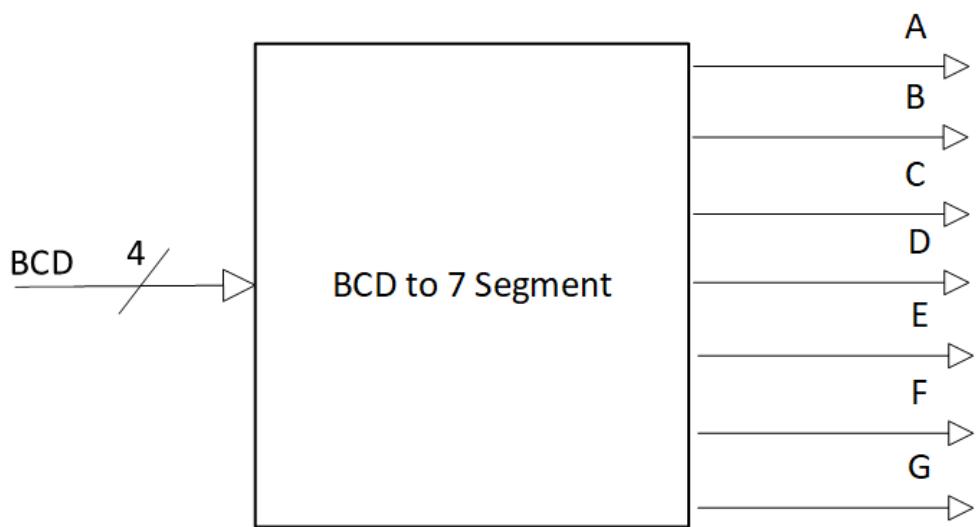
BCD-TO-7-SEGMENT CODE CONVERTER

Rangkaian ini digunakan untuk mengkonversikan suatu nilai desimal terkode biner (BCD) ke pola segmen yang sesuai pada display 7-segmen. Karena nilai BCD adalah angka 4-bit pada jangkauan 0-9, bagaimana kita memperlakukan nilai 10-15(don't care atau tidak) akan berpengaruh pada desain kita.

7-segmen biasanya diidentifikasi dalam industri menggunakan huruf a-g seperti pada Gambar 2 berikut ini:

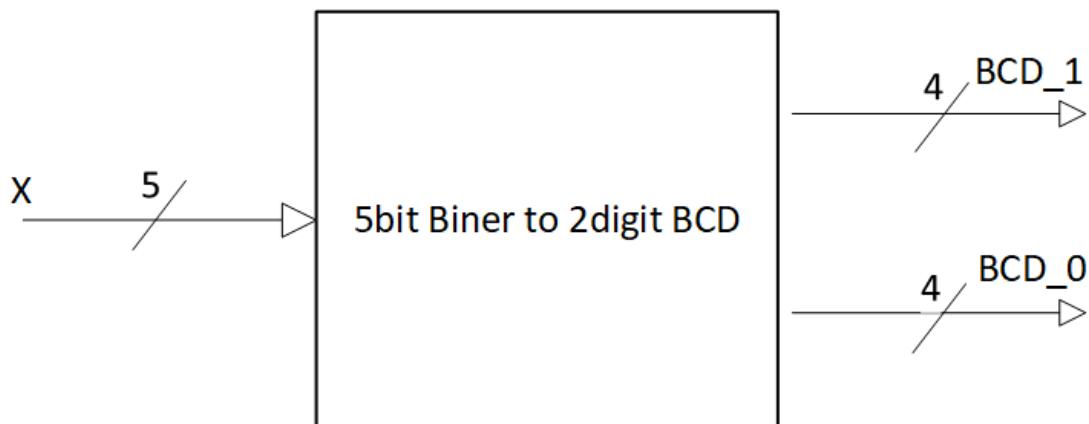


Gambar 2. (a) Konvensi penomoran 7-segmen, (b) Pola Display 7-segmen



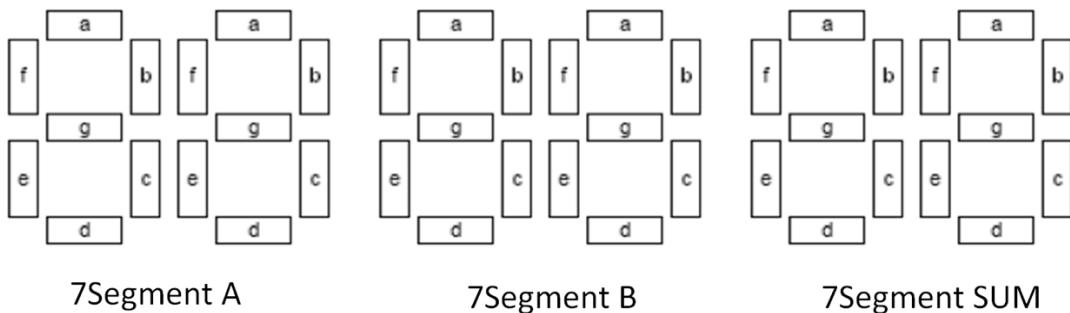
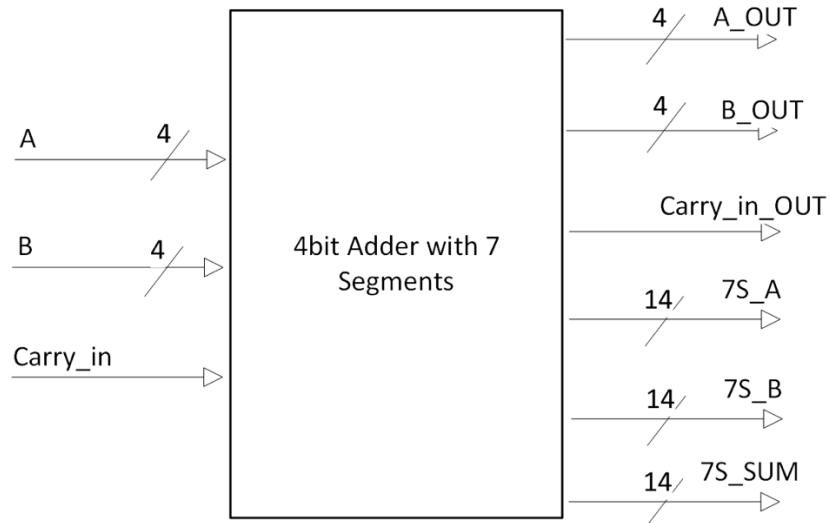
5BIT BINARY-TO-BCD CODE CONVERTER

Rangkaian ini digunakan untuk mengkonversikan suatu nilai biner 5bit ke BCD. Biner 5 bit bernilai dari 0 – 31. Untuk itu dibutuhkan 2 digit BCD (satuan dan puluhan).



4BIT ADDER WITH 7 SEGMENT

Rangkaian ini berfungsi untuk melakukan penjumlahah 2 buah bilangan biner 4 bit dan 1 bit carry. Rangkaian ini juga menampilkan nilai kedua buah bilangan dan hasil penjumlahan pada 7 segment. Setiap bilangan dan hasil penjumlahan ditampilkan pada 2 buah 7 segment (satuan dan puluhan).



1.4 TUGAS PENDAHULUAN

1. Rancanglah BCD-to-7Segment decoder dalam bentuk VHDL dan pastikan sudah benar!
2. Rancanglah 5bitBiner-to-2BCD decoder dalam bentuk VHDL dan pastikan sudah benar!
3. Rancanglah 4bitAdder dalam bentuk VHDL dan pastikan sudah benar!
4. Buatlah test vector untuk percobaan 2C dan gunakan tabel seperti di Apendiks! Test vector ini merupakan kombinasi nilai A, B, dan Carry. Minimal jumlah test vector adalah 64 buah dengan kombinasi yang Anda dapat tentukan sendiri. Kombinasi nilai A, B, dan Carry tidak boleh berulang.

Semua rangkaian dirancang pada Logic Gate Level (menggunakan SOP/POS).

Semua VHDL diberikan *comment*.

Semua VHDL file harus dibawa untuk pelaksanaan praktikum.

1.5 PERCOBAAN

PERALATAN YANG DIGUNAKAN

- Board FPGA tipe DE10-Lite
- Catu daya + kabel dan konektor tambahan serta kabel *downloader*
- Osiloskop

PERCOBAAN 2A: IMPLEMENTASI BCD-TO-7SEGMENT PADA FPGA BOARD

Pada percobaan ini Anda akan melakukan implementasi dari hasil rancangan BCD-to-7Segment yang telah Anda buat pada tugas pendahuluan.

PROSEDUR PERCOBAAN:

1. Buatlah folder baru untuk melakukan percobaan pada praktikum ini. Folder ini nantinya digunakan sebagai direktori kerja, untuk menyimpan file-file yang berhubungan dengan praktikum ini.
2. Buatlah project baru dan import file VHDL BCD-to-7Segment yang telah Anda buat pada tugas pendahuluan.
3. Lakukan pemetaan pin input dan output. Input menggunakan slide switch dan output menggunakan 7 segment. Buat tabel pin plannernya! Untuk informasi lebih lengkap, lihat Apendiks.
4. Implementasikan pada FPGA board.
5. Mainkan switch dan perhatikan apakah program sudah berjalan dengan benar. Gunakan table pengujian seperti pada Apendiks.
6. Cata hasil percobaan di BCL.

PERCOBAAN 2B: MERANCANG BCD-TO-7SEGMENT DENGAN LEVEL ABSTRAKSI BEHAVIORAL

Pada percobaan kali ini kita akan mengimplementasikan desain dengan level abstraksi yang lebih tinggi. Level abstraksi yang tinggi artinya lebih dekat dengan cara manusia berpikir. Pada percobaan ini ditunjukkan bahwa kita sering kali tidak perlu melakukan/mencari persamaan logika untuk setiap signal/variable. Pada contoh ini, praktikan cukup menentukan bentuk keluaran, untuk setiap jenis input yang diinginkan. Proses merubah menjadi persamaan Boolean, meminimisasi, dan membuat rangkaian gerbang logikanya dikerjakan oleh tool/software. Dengan cara ini manusia/engineer dapat membuat rangkaian yang lebih besar/kompleks karena tidak perlu memikirkan detailnya.

PROSEDUR PERCOBAAN:

1. Buatlah folder baru untuk melakukan percobaan pada praktikum ini. Folder ini nantinya digunakan sebagai direktori kerja, untuk menyimpan file-file yang berhubungan dengan praktikum ini.
2. Buatlah file **DUT (Device Under Test)** dengan cara mengetikkan *script* di bawah ini menggunakan *text editor*, kemudian simpan file tersebut di folder yang telah dibuat pada langkah sebelumnya.

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.numeric_std.all;

ENTITY bcd IS PORT (
    SW : IN STD_LOGIC_VECTOR (3 DOWNTO 0);
    HEX1 : OUT STD_LOGIC_VECTOR (1 TO 7));
```

```

END bcd;

ARCHITECTURE behavioral OF bcd IS

    CONSTANT NOL      : STD_LOGIC_VECTOR(3 DOWNTO 0) := "0000";
    CONSTANT SATU     : STD_LOGIC_VECTOR(3 DOWNTO 0) := "0001";
    CONSTANT DUA      : STD_LOGIC_VECTOR(3 DOWNTO 0) := "0010";
    CONSTANT TIGA     : STD_LOGIC_VECTOR(3 DOWNTO 0) := "0011";
    CONSTANT EMPAT    : STD_LOGIC_VECTOR(3 DOWNTO 0) := "0100";
    CONSTANT LIMA      : STD_LOGIC_VECTOR(3 DOWNTO 0) := "0101";
    CONSTANT ENAM     : STD_LOGIC_VECTOR(3 DOWNTO 0) := "0110";
    CONSTANT TUJUH    : STD_LOGIC_VECTOR(3 DOWNTO 0) := "0111";
    CONSTANT DELAPAN   : STD_LOGIC_VECTOR(3 DOWNTO 0) := "1000";
    CONSTANT SEMBILAN  : STD_LOGIC_VECTOR(3 DOWNTO 0) := "1001";

BEGIN

    PROCESS (SW)
    BEGIN
        CASE SW IS
            WHEN NOL      => HEX1 <= "1111110";
            WHEN SATU     => HEX1 <= "0110000";
            WHEN DUA      => HEX1 <= "1101101";
            WHEN TIGA     => HEX1 <= "1111001";
            WHEN EMPAT    => HEX1 <= "0110011";
            WHEN LIMA      => HEX1 <= "1011011";
            WHEN ENAM     => HEX1 <= "1011111";
            WHEN TUJUH    => HEX1 <= "1110000";
            WHEN DELAPAN   => HEX1 <= "1111111";
            WHEN SEMBILAN  => HEX1 <= "1110011";
            WHEN OTHERS  => HEX1 <= "0000000";
        END CASE;
    END PROCESS;

END behavioral;

```

3. Buatlah file **Testbench** dengan cara mengetikkan *script* di bawah ini menggunakan *text editor*, kemudian simpan file tersebut di folder yang telah dibuat pada langkah sebelumnya.

```

LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_unsigned.all;

ENTITY tb_bcd IS
END tb_bcd;

ARCHITECTURE behavioral OF tb_bcd IS

    SIGNAL clk  : STD_LOGIC := '0';
    SIGNAL SW   : STD_LOGIC_VECTOR (3 DOWNTO 0) := "0000";
    SIGNAL HEX1 : STD_LOGIC_VECTOR (1 TO 7);

    COMPONENT bcd IS
        PORT(  SW   : IN STD_LOGIC_VECTOR (3 DOWNTO 0);
                 HEX1 : OUT STD_LOGIC_VECTOR (1 TO 7));
    ENDCOMPONENT;

BEGIN

```

```

dut : bcd
PORT MAP (
    SW      => SW      ,
    HEX1   => HEX1  );

clock : PROCESS
BEGIN
WAIT FOR 50 ps; clk  <= not clk;
end PROCESS clock;

increment: PROCESS (clk)
BEGIN
IF (clk'EVENT AND clk = '1') THEN
    SW <= SW + "0001";
END IF;
END PROCESS;

END behavioral;

```

4. Buatlah file **sim.do** dengan cara mengetikkan script seperti yang telah dilakukan pada **modul latihan**. Lakukan beberapa modifikasi penyesuaian pada *script* tersebut.
5. Pada jendela *transcript* ketikkan “do sim.do” untuk menjalankan simulasi. Ambil gambar sinyal hasil simulasi tersebut, kemudian sertakan dalam laporan ! Analisis sinyal hasil simulasi tersebut !
6. Lakukan pemetaan pin input dan output. Input menggunakan slide switch dan output menggunakan 7 segment. Buat tabel pin plannernya! Untuk informasi lebih lengkap, lihat Apendiks.
7. Implementasikan pada FPGA board.
8. Mainkan switch dan perhatikan apakah program sudah berjalan dengan benar. Gunakan table pengujian seperti pada Apendiks.
9. Cata hasil percobaan di BCL.

PERCOBAAN 2C: MEMBUAT RANGKAIAN 4BIT ADDER WITH 7 SEGMENT

Spesifikasi rangkaian sebagai berikut:

Tabel Spesifikasi

		PIN IN/OUT
Input	A (4 bit) B (4 bit) Carry_in (1 bit)	Slide switch [3-0] Slide switch [7-4] Slide switch [8]
Output	A_out (4 bit) B_out (4 bit) Carry_in_out (1 bit) 7Segment_A0 (7 bit)	LED [3-0] LED [7-4] LED [8] HEX4

	7Segment_A1 (7 bit)	HEX5
	7Segment_B0 (7 bit)	HEX3
	7Segment_B1 (7 bit)	HEX2
	7Segment_S0 (7 bit)	HEX1
	7Segment_S1 (7 bit)	HEX0
Fungsi	Menjumlahkan 2 buah unsigned integer 4 bit dan carry 1 bit. Menampilkan nilai 2 buah bilangan dan hasil penjumlahan dalam decimal. Menampilkan flag nilai carry (0 atau 1).	

PEMBUATAN PROJECT.

1. Buatlah project Quartus baru bernama **4bitAdder7S** pada direktori **4bitAdder7S**
2. Import file VHDL yang telah dibuat pada tugas pendahuluan sebelumnya.

PEMBUATAN VHDL

1. Desainlah sebuah rangkaian 4-bit Adder with 7 Segment seperti yang dispesifikasikan dengan memanfaatkan VHDL yang telah Anda buat pada tugas pendahuluan!
2. Lakukan pin planner sesuai dengan tabel spesifikasi.
3. Memprogram kedalam FPGA
 - a. Coba Anda download program 4bitAdder7S Anda kedalam board FPGA yang tersedia, lihat kembali modul latihan untuk cara pemrograman.
 - b. Mainkan 9 switch yang kita pakai pada percobaan ini dan lihat apakah program kita sudah berjalan dengan benar. Gunakan tabel pengujian seperti pada Apendiks. Minimal jumlah test vector adalah 64 buah dengan kombinasi yang Anda dapat tentukan sendiri. Kombinasi nilai A, B, dan Carry tidak boleh berulang.
 - c. Catat Hasil percobaan pada BCL Anda.

Setelah menyelesaikan ini, simpan seluruh file percobaan 2 karena **akan digunakan kembali** pada modul 3 dan modul 4. Jika belum selesai maka selesaikan di rumah.

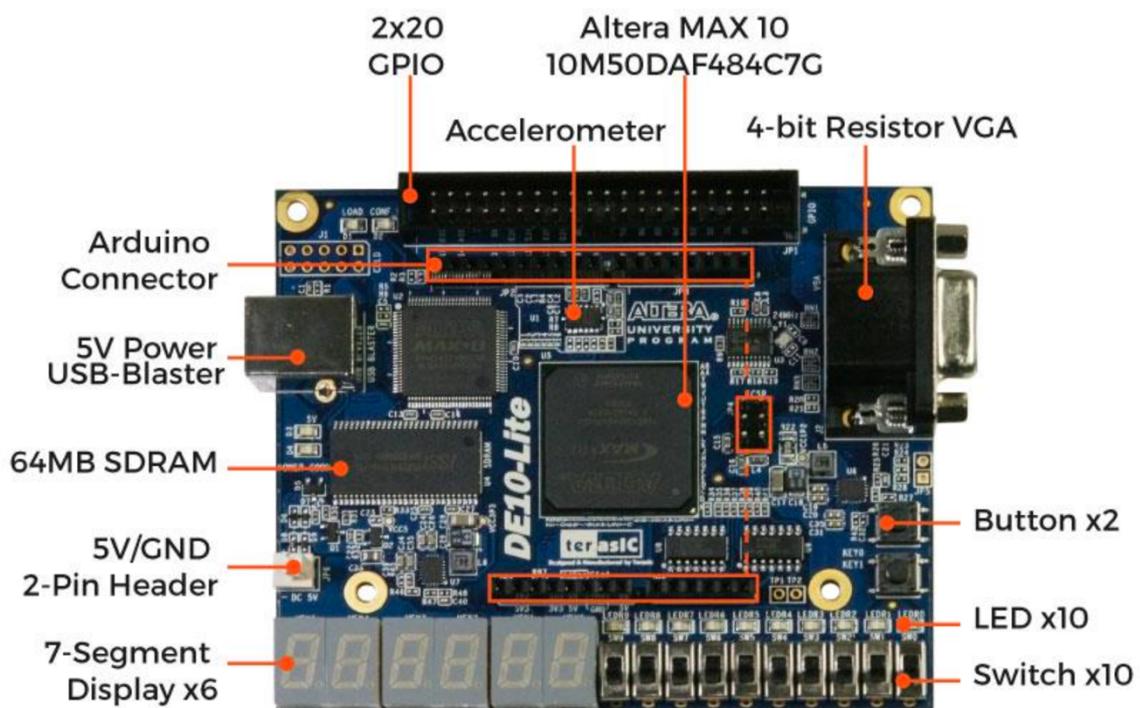
1.6 MENGAKHIRI PERCOBAAN

Prosedur untuk mengakhiri percobaan:

1. Sebelum keluar dari ruang praktikum, rapikan meja praktikum. **Rapikan kabel dan matikan komputer, osiloskop, generator sinyal, dan power supply DC.** Cabut daya dari jala-jala ke kit FPGA dan letakkan kembali pada tempat semula.
2. Periksa lagi lembar penggunaan meja. Praktikan yang tidak menandatangani **lembar penggunaan meja** atau merapikan meja ketika praktikum berakhir akan mendapatkan **potongan nilai sebesar minimal 10**.
3. **Pastikan asisten telah menandatangani catatan percobaan kali ini** pada Buku Catatan Laboratorium Anda. Catatan percobaan yang tidak ditandatangani oleh asisten tidak akan dinilai.

APENDIKS

DE10-LITE FPGA BOARD



PIN ASSIGNMENT OF SLIDE SWITCHES



Figure 3-15 Connections between the slide switches and MAX 10 FPGA

Table 3-4 Pin Assignment of Slide Switches

Signal Name	FPGA Pin No.	Description	I/O Standard
SW0	PIN_C10	Slide Switch[0]	3.3-V LVTTL
SW1	PIN_C11	Slide Switch[1]	3.3-V LVTTL
SW2	PIN_D12	Slide Switch[2]	3.3-V LVTTL
SW3	PIN_C12	Slide Switch[3]	3.3-V LVTTL
SW4	PIN_A12	Slide Switch[4]	3.3-V LVTTL
SW5	PIN_B12	Slide Switch[5]	3.3-V LVTTL
SW6	PIN_A13	Slide Switch[6]	3.3-V LVTTL
SW7	PIN_A14	Slide Switch[7]	3.3-V LVTTL
SW8	PIN_B14	Slide Switch[8]	3.3-V LVTTL
SW9	PIN_F15	Slide Switch[9]	3.3-V LVTTL

PIN ASSIGNMENT OF LEDS

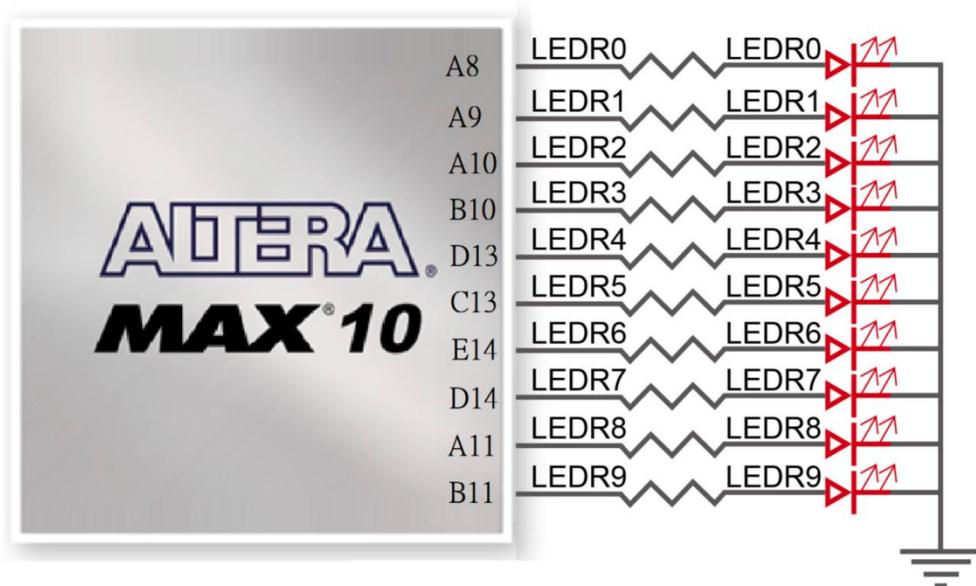


Figure 3-16 Connections between the LEDs and MAX 10 FPGA

Table 3-5 Pin Assignment of LEDs

Signal Name	FPGA Pin No.	Description	I/O Standard
LEDR0	PIN_A8	LED [0]	3.3-V LVTTL
LEDR1	PIN_A9	LED [1]	3.3-V LVTTL
LEDR2	PIN_A10	LED [2]	3.3-V LVTTL
LEDR3	PIN_B10	LED [3]	3.3-V LVTTL
LEDR4	PIN_D13	LED [4]	3.3-V LVTTL
LEDR5	PIN_C13	LED [5]	3.3-V LVTTL
LEDR6	PIN_E14	LED [6]	3.3-V LVTTL
LEDR7	PIN_D14	LED [7]	3.3-V LVTTL
LEDR8	PIN_A11	LED [8]	3.3-V LVTTL
LEDR9	PIN_B11	LED [9]	3.3-V LVTTL

PIN ASSIGNMENT OF 7-SEGMENT DISPLAYS

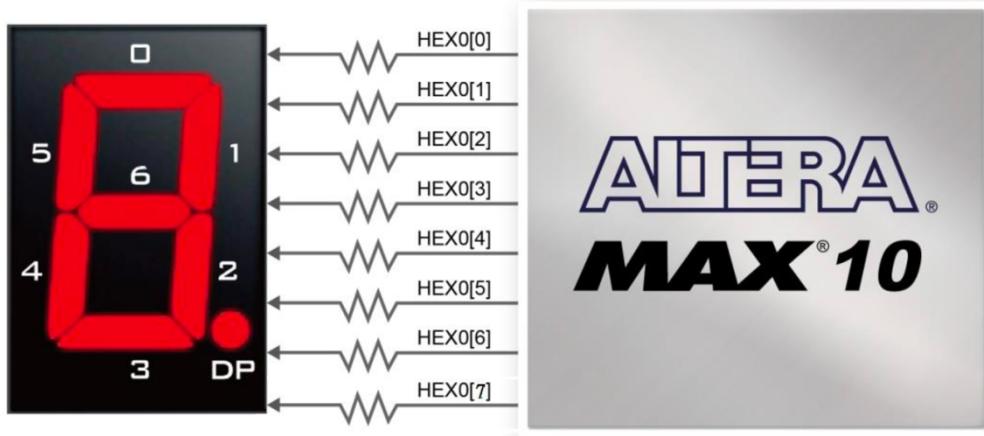


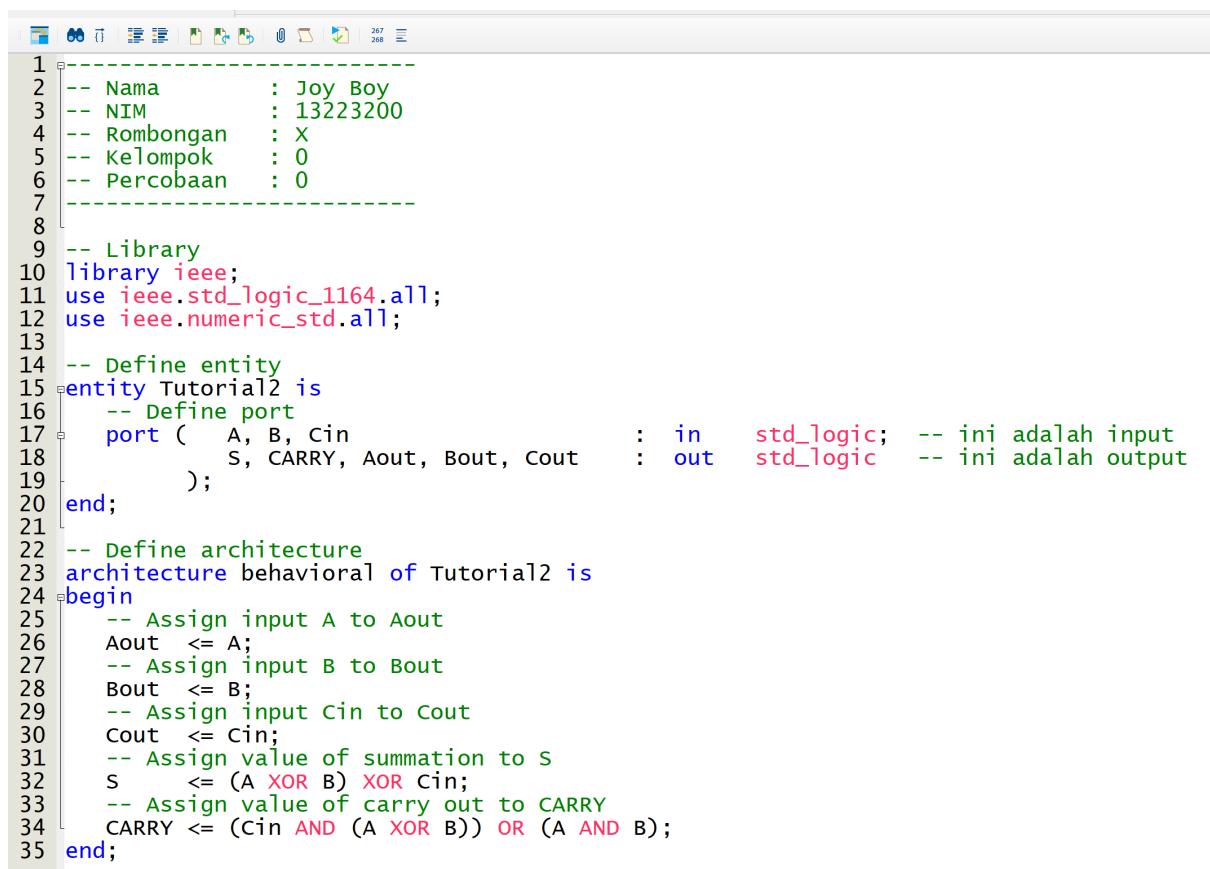
Figure 3-17 Connections between the 7-segment display HEX0 and the MAX 10 FPGA

Table 3-6 Pin Assignment of 7-segment Displays

Signal Name	FPGA Pin No.	Description	I/O Standard
HEX00	PIN_C14	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX01	PIN_E15	Seven Segment Digit 0[1]	3.3-V LVTTL
HEX02	PIN_C15	Seven Segment Digit 0[2]	3.3-V LVTTL
HEX03	PIN_C16	Seven Segment Digit 0[3]	3.3-V LVTTL
HEX04	PIN_E16	Seven Segment Digit 0[4]	3.3-V LVTTL
HEX05	PIN_D17	Seven Segment Digit 0[5]	3.3-V LVTTL
HEX06	PIN_C17	Seven Segment Digit 0[6]	3.3-V LVTTL
HEX07	PIN_D15	Seven Segment Digit 0[7], DP	3.3-V LVTTL
HEX10	PIN_C18	Seven Segment Digit 1[0]	3.3-V LVTTL
HEX11	PIN_D18	Seven Segment Digit 1[1]	3.3-V LVTTL
HEX12	PIN_E18	Seven Segment Digit 1[2]	3.3-V LVTTL
HEX13	PIN_B16	Seven Segment Digit 1[3]	3.3-V LVTTL

HEX14	PIN_A17	Seven Segment Digit 1[4]	3.3-V LVTTL
HEX15	PIN_A18	Seven Segment Digit 1[5]	3.3-V LVTTL
HEX16	PIN_B17	Seven Segment Digit 1[6]	3.3-V LVTTL
HEX17	PIN_A16	Seven Segment Digit 1[7] , DP	3.3-V LVTTL
HEX20	PIN_B20	Seven Segment Digit 2[0]	3.3-V LVTTL
HEX21	PIN_A20	Seven Segment Digit 2[1]	3.3-V LVTTL
HEX22	PIN_B19	Seven Segment Digit 2[2]	3.3-V LVTTL
HEX23	PIN_A21	Seven Segment Digit 2[3]	3.3-V LVTTL
HEX24	PIN_B21	Seven Segment Digit 2[4]	3.3-V LVTTL
HEX25	PIN_C22	Seven Segment Digit 2[5]	3.3-V LVTTL
HEX26	PIN_B22	Seven Segment Digit 2[6]	3.3-V LVTTL
HEX27	PIN_A19	Seven Segment Digit 2[7] , DP	3.3-V LVTTL
HEX30	PIN_F21	Seven Segment Digit 3[0]	3.3-V LVTTL
HEX31	PIN_E22	Seven Segment Digit 3[1]	3.3-V LVTTL
HEX32	PIN_E21	Seven Segment Digit 3[2]	3.3-V LVTTL
HEX33	PIN_C19	Seven Segment Digit 3[3]	3.3-V LVTTL
HEX34	PIN_C20	Seven Segment Digit 3[4]	3.3-V LVTTL
HEX35	PIN_D19	Seven Segment Digit 3[5]	3.3-V LVTTL
HEX36	PIN_E17	Seven Segment Digit 3[6]	3.3-V LVTTL
HEX37	PIN_D22	Seven Segment Digit 3[7] , DP	3.3-V LVTTL
HEX40	PIN_F18	Seven Segment Digit 4[0]	3.3-V LVTTL
HEX41	PIN_E20	Seven Segment Digit 4[1]	3.3-V LVTTL
HEX42	PIN_E19	Seven Segment Digit 4[2]	3.3-V LVTTL
HEX43	PIN_J18	Seven Segment Digit 4[3]	3.3-V LVTTL
HEX44	PIN_H19	Seven Segment Digit 4[4]	3.3-V LVTTL
HEX45	PIN_F19	Seven Segment Digit 4[5]	3.3-V LVTTL
HEX46	PIN_F20	Seven Segment Digit 4[6]	3.3-V LVTTL
HEX47	PIN_F17	Seven Segment Digit 4[7] , DP	3.3-V LVTTL
HEX50	PIN_J20	Seven Segment Digit 5[0]	3.3-V LVTTL
HEX51	PIN_K20	Seven Segment Digit 5[1]	3.3-V LVTTL
HEX52	PIN_L18	Seven Segment Digit 5[2]	3.3-V LVTTL
HEX53	PIN_N18	Seven Segment Digit 5[3]	3.3-V LVTTL
HEX54	PIN_M20	Seven Segment Digit 5[4]	3.3-V LVTTL
HEX55	PIN_N19	Seven Segment Digit 5[5]	3.3-V LVTTL
HEX56	PIN_N20	Seven Segment Digit 5[6]	3.3-V LVTTL
HEX57	PIN_L19	Seven Segment Digit 5[7] , DP	3.3-V LVTTL

CONTOH PEMBERIAN COMMENT PADA VHDL



```
1  -- Nama      : Joy Boy
2  -- NIM       : 13223200
3  -- Rombongan : X
4  -- Kelompok  : 0
5  -- Percobaan : 0
6  -----
7
8
9  -- Library
10 library ieee;
11 use ieee.std_logic_1164.all;
12 use ieee.numeric_std.all;
13
14 -- Define entity
15 entity Tutorial2 is
16   -- Define port
17   port ( A, B, Cin      : in  std_logic;  -- ini adalah input
18          S, CARRY, Aout, Bout, Cout    : out std_logic); -- ini adalah output
19
20 end;
21
22 -- Define architecture
23 architecture behavioral of Tutorial2 is
24 begin
25   -- Assign input A to Aout
26   Aout  <= A;
27   -- Assign input B to Bout
28   Bout  <= B;
29   -- Assign input Cin to Cout
30   Cout  <= Cin;
31   -- Assign value of summation to S
32   S     <= (A XOR B) XOR Cin;
33   -- Assign value of carry out to CARRY
34   CARRY <= (Cin AND (A XOR B)) OR (A AND B);
35 end;
```

TABEL PENGUJIAN BCD-TO-7SEGMENT (PERCOBAAN 2A DAN 2B)

Catatan: kolom “tampilan 7segment” digambarkan pola 7 segment yang menyala.

BCD3	BCD2	BCD1	BCD0	Tampilan 7Segment
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

TABEL PENGUJIAN 4BIT ADDER WITH 7 SEGMENT

Catatan: kolom “tampilan 7segment” digambarkan pola 7 segment yang menyala.

Test Vector #	CARRY	A	B	Tampilan 7 Segment				
				7S A1	7S A0	7S B1	7S B0	7S SUM1
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								
25								

26									
27									
28									
29									
30									
31									
32									
33									
34									
35									
36									
37									
38									
39									
40									
41									
42									
43									
44									
45									
46									
47									
48									
49									
50									
51									
52									
53									
54									
55									

56									
57									
58									
59									
60									
61									
62									
63									
64									