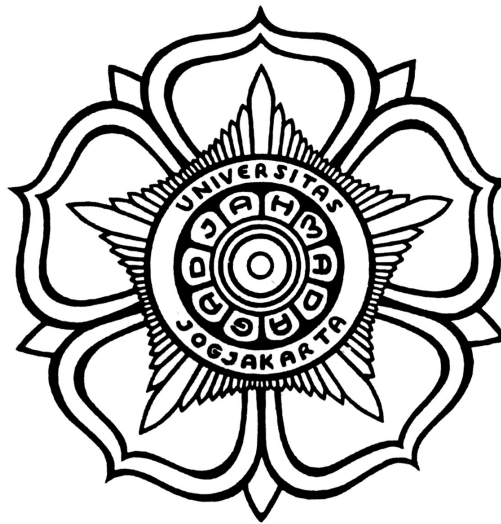


MODUL 07

PRAKTIKUM SISTEM DIGITAL

TNF 2178 - 1 SKS



Disusun oleh:

Prof. Ir. Sunarno, M.Eng., Ph.D.

dan

Tim Asisten Praktikum Sistem Digital

**LABORATORIUM SENSOR DAN SISTEM TELEKONTROL
DEPARTEMEN TEKNIK NUKLIR DAN TEKNIK FISIKA
FAKULTAS TEKNIK
UNIVERSITAS GADJAH MADA
2018**

Daftar Isi

1	COUNTER & SEVEN SEGMENT	5
1.1	Tujuan	5
1.2	Materi	5
1.3	Teori	5
1.3.1	Counter	5
1.3.2	Register	6
1.3.3	Ripple Counter	8
1.3.4	Synchronous Counter	10
1.3.5	Perancangan Counter	11
1.4	Alat dan bahan	12
1.5	Metodologi Percobaan	12
1.5.1	Counter	12
1.5.2	Seven segment	14

Chapter 1

COUNTER & SEVEN SEGMENT

1.1 Tujuan

- a. Mahasiswa mengenal cara kerja *counter* dan *seven segment*.
- b. Mahasiswa dapat merangkai pencacah maju dan mampu menampilkan pada *display seven segment*.

1.2 Materi

- a. Counter up
- b. Seven segment

1.3 Teori

1.3.1 Counter

Counter juga disebut pencacah atau penghitung yaitu rangkaian logika sekuensial yang digunakan untuk menghitung jumlah pulsa yang diberikan pada bagian masukan. *Counter* digunakan untuk berbagai operasi aritmatika, pembagi frekuensi, penghitung jarak (odometer), penghitung kecepatan (speedometer), yang pengembangannya digunakan luas dalam aplikasi perhitungan pada instrumen ilmiah, kontrol industri, komputer, perlengkapan komunikasi, dan sebagainya.

Counter tersusun atas sederetan flip-flop yang dimanipulasi sedemikian rupa dengan menggunakan peta Karnough sehingga pulsa yang masuk dapat dihitung sesuai rancangan. Dalam perancangannya *counter* dapat tersusun atas semua jenis flip-flop, tergantung karakteristik masing-masing flip-flop tersebut. Secara global *counter* terbagi atas 2 jenis, yaitu :

1. *Asynchronous Counter*

Seperti tersebut pada bagian sebelumnya *Asynchronous counter* tersusun atas *flip-flop* yang dihubungkan seri dan pemicuannya tergantung dari *flip-flop* sebelumnya, kemudian menjalar sampai *flip-flop* MSB-nya. Karena itulah *Asynchronous counter* sering disebut juga sebagai *ripple-through counter*.

2. *Synchronous Counter*

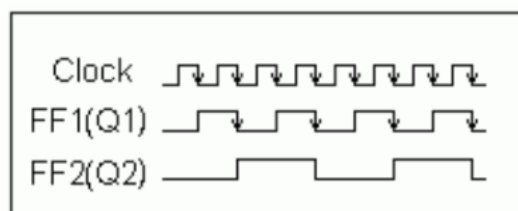
Synchronous counter memiliki pemicuan dari sumber *clock* yang sama dan susunan *flip-flop*-nya adalah paralel. Dalam *synchronous counter* ini sendiri terdapat perbedaan penempatan atau manipulasi gerbang dasarnya yang menyebabkan perbedaan waktu tunda yang disebut *carry propagation delay*.

Dengan memanipulasi koneksi *flip-flop* berdasarkan peta Karnaugh atau *timing diagram* dapat dihasilkan *counter* acak, *shift counter* (*counter* sebagai fungsi register) atau juga *up-down counter*. Prosedur desain suatu *synchronous counter* sama seperti pada prosedur desain *flip-flop* (pertemuan IV).

Penerapan *counter* dalam aplikasinya adalah berupa chip IC baik IC TTL, maupun CMOS, antara lain adalah: (TTL) 7490, 7493, 74190, 74191, 74192, 74193, (CMOS) 4017, 4029, 4042, dan lain-lain.

Counter Sebagai Pembagi Frekuensi

Counter sebagai pembagi frekuensi merupakan salah satu aplikasi dari *counter*. Dalam rangkaian ini, frekuensi keluaran suatu *flip-flop* merupakan setengah frekuensi masukannya. Untuk lebih jelasnya, dapat diamati dalam diagram waktu di bawah ini. Pada gambar tersebut dapat dilihat frekuensi masukan



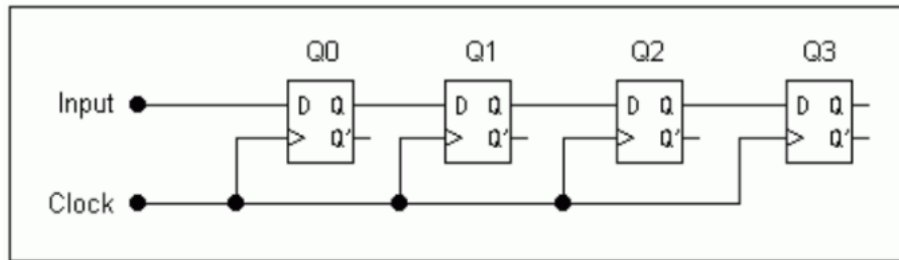
FF1 (berasal dari *clock*) dua kali lipat frekuensi Q1.

1.3.2 Register

Register merupakan sekelompok *flip-flop* yang dapat menyimpan informasi biner yang terdiri dari bit majemuk. Register dengan n *flip-flop* mampu menyimpan sebesar n bit. Ada dua cara untuk menyimpan dan membaca data ke dalam register, yaitu **seri** dan **paralel**. Dalam operasi paralel, penyimpanan atau pembacaan dilakukan secara serentak oleh semua tingkat register. Sedangkan untuk operasi seri, diterapkan secara *sequential* bit demi bit sampai semua tingkat register terpenuhi.

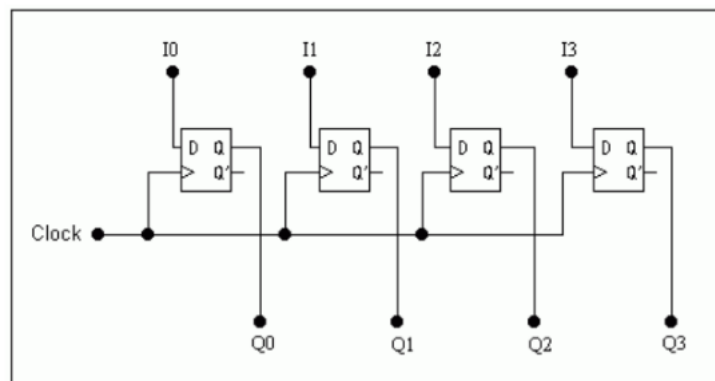
1. Register Serial In-Serial Out

Pada Register Serial In-Serial Out, jalur masuk data berjumlah satu dan jalur keluarannya juga berjumlah satu. Pada jenis register ini data mengalami pergeseran, *flip-flop* pertama menerima masukan dari *input*, sedangkan *flip-flop* kedua menerima masukan dari *flip-flop* pertama, dan seterusnya.



2. Register Paralel In-Paralel Out

Register Paralel In-Paralel Out mempunyai jalur masukan dan keluaran sesuai dengan jumlah *flip-flop* yang menyusunnya. Pada register jenis ini, data masuk dan keluar secara serentak. Dan hanya membutuhkan satu kali picu.

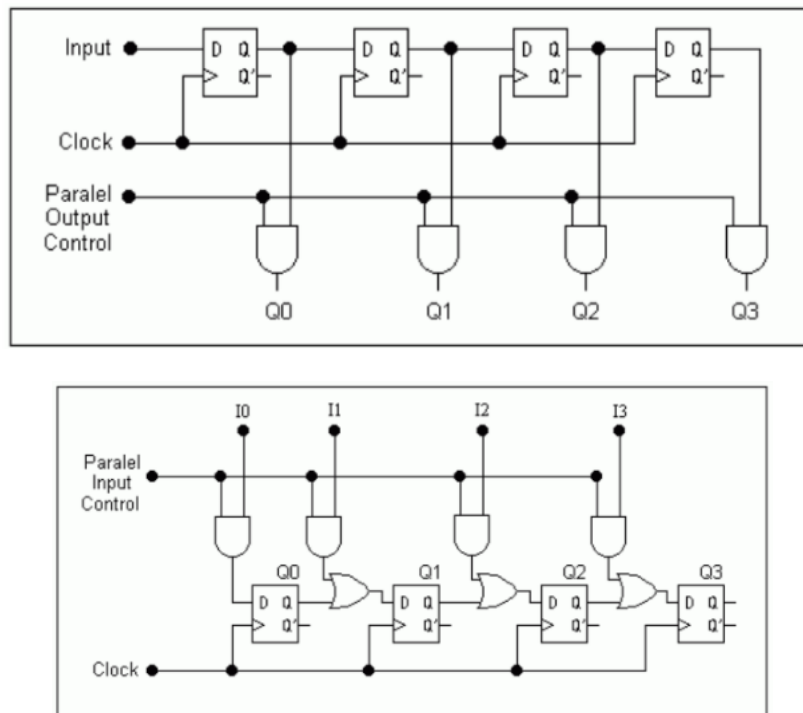


3. Register Serial In-Paralel Out

Register serial In-Paralel Out mempunyai satu saluran masukan dan saluran keluaran sejumlah *flip-flop* yang menyusunnya. Data masuk satu-persatu (secara serial) dan dikeluarkan secara serentak. Pengeluaran data dikendalikan oleh sebuah sinyal kontrol. Selama sinyal kontrol tidak diberikan, data akan tetap tersimpan dalam register.

4. Register Paralel In-Serial Out

Register Paralel In-Serial Out mempunyai jalur masukan sesuai dengan jumlah *flip-flop* yang menyusunnya, dan hanya mempunyai satu jalur keluaran. Data masuk ke dalam register secara serentak dengan dikendalikan sinyal kontrol, sedangkan data keluar satu-persatu (secara serial).



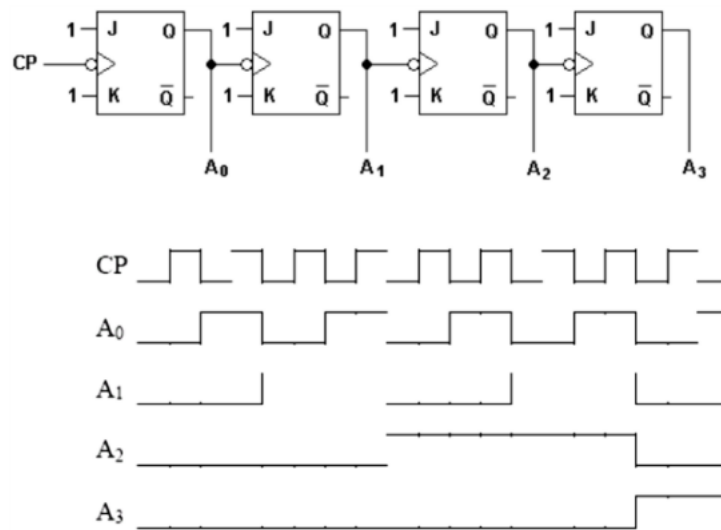
1.3.3 Ripple Counter

Selain register digunakan untuk transfer data, register juga mempunyai kegunaan lain yaitu sebagai *counter* (pencacah). *Counter* dapat dibedakan menjadi dua macam, yang pertama adalah *Ripple counter* dimana *output flip-flop* yang satu digunakan sebagai *clock* pulsa pada *flip-flop* lainnya. Sedangkan *synchronous counter* semua *flip-flop* mempunyai *clock* pulsa yang sama.

1. Binary Ripple Counter

Binary ripple counter terdiri dari *flip-flop complement* (T *flip-flop* dan JK *flip-flop*) yang dirangkai secara seri dengan setiap *output flip-flop* dihubungkan pada *input CP flip-flop* yang mempunyai urutan lebih tinggi.

A_3	A_2	A_1	A_0
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



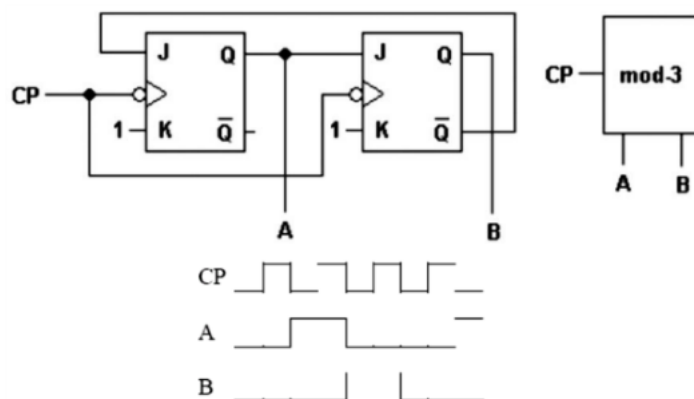
2. Modulus Counter

Rangkaian *counter-counter* diatas hanya akan kembali ke keadaan semula (reset) dalam $2n$ pulsa. Apabila kita menginginkan suatu *flip-flop* akan reset setelah hitungan tertentu (selain hitungan $2n$) maka *counter* ini disebut sebagai modulus *counter*.

3. Counter Mod-3

Berikut ini adalah *counter* mod-3 dengan menggunakan JK *flip-flop*.

B	A	count
0	0	0
0	1	1
1	0	2
0	0	0



4. Counter Mod-6

Dasar dari *flip-flop* adalah *counter* mod-2. Sehingga jika kita ingin mem-

1.3.5 Perancangan Counter

Perancangan *counter* dapat dibagi menjadi 2, yaitu dengan menggunakan peta Karnaugh, dan dengan diagram waktu. Berikut ini akan dijelaskan langkah-langkah dalam merancang suatu *counter*.

1. Perancangan Counter Menggunakan Peta Karnaugh

Umumnya perancangan dengan peta Karnaugh ini digunakan dalam merancang *synchronous counter*. Langkah-langkah perancangannya:

- a) Dengan mengetahui urutan keluaran *counter* yang akan dirancang, kita tentukan masukan masing-masing *flip-flop* untuk setiap kondisi keluaran, dengan menggunakan tabel kebalikan.
- b) Cari fungsi boolean masing-masing masukan *flip-flop* dengan menggunakan peta Karnaugh. Usahakan untuk mendapatkan fungsi yang sesederhana mungkin, agar rangkaian *counter* menjadi sederhana.
- c) Buat rangkaian *counter*, dengan fungsi masukan *flip-flop* yang telah ditentukan. Pada umumnya digunakan gerbang-gerbang logika untuk membentuk fungsi tersebut.

2. Perancangan Counter Menggunakan Diagram Waktu

Umumnya perancangan dengan diagram waktu digunakan dalam merancang *asynchronous counter*, karena kita dapat mengamati dan menentukan sumber pemicuan suatu *flip-flop* dari *flip-flop* lainnya. Adapun langkah-langkah perancangannya:

- a) Menggambarkan diagram waktu *clock*, tentukan jenis pemicuan yang digunakan, dan keluaran masing-masing *flip-flop* yang kita inginkan. Untuk n kondisi keluaran, terdapat n jumlah pulsa *clock*.
- b) Dengan melihat keluaran masing-masing *flip-flop* sebelum dan sesudah *clock* aktif (Q_n dan Q_{n+1}), tentukan fungsi masukan *flip-flop* dengan menggunakan tabel kebalikan.
- c) Menggambarkan fungsi masukan tersebut pada diagram waktu yang sama.
- d) Sederhanakan fungsi masukan yang telah diperoleh sebelumnya, dengan melihat kondisi logika dan kondisi keluaran *flip-flop*. Untuk *flip-flop* R-S dan J-K kondisi *don't care* (x) dapat dianggap sama dengan 0 atau 1.
- e) Tentukan (minimal satu) *flip-flop* yang dipicu oleh keluaran *flip-flop* lain. Hal ini dapat dilakukan dengan mengamati perubahan keluaran suatu *flip-flop* setiap perubahan keluaran *flip-flop* lain, sesuai dengan jenis pemicuannya.
- f) Buat rangkaian *counter*, dengan fungsi masukan *flip-flop* yang telah ditentukan. Pada umumnya digunakan gerbang-gerbang logika untuk membentuk fungsi tersebut.

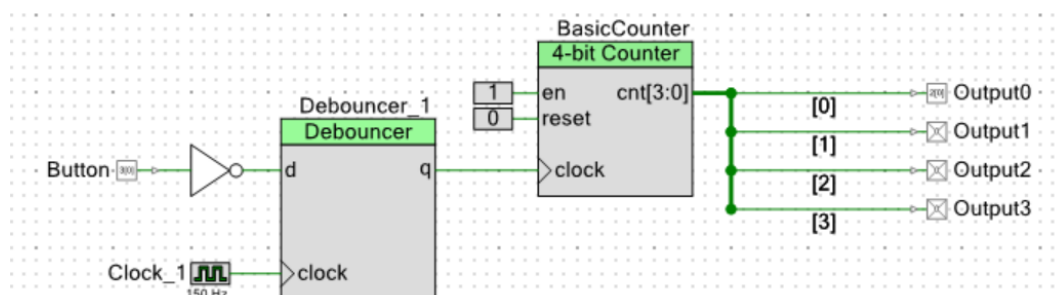
1.4 Alat dan bahan

- Buku praktikum
- Modul PSoC
- Kabel jumper
- Laptop

1.5 Metodologi Percobaan

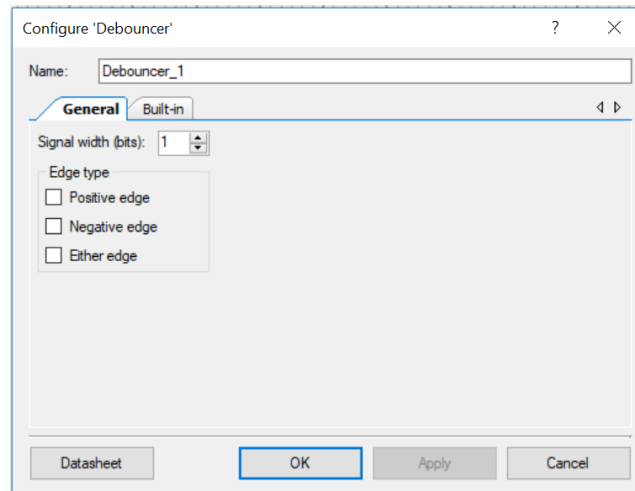
1.5.1 Counter

- Siapkan alat dan bahan yang dibutuhkan.
- Nyalakan laptop, kemudian buka perangkat lunak PSoC Creator.
- Buat *Project* baru dengan nama *workspace* Modul_5_Counter.
- Buat rangkaian pada *worksheet* seperti pada Gambar 1.1. Blok-blok yang dibutuhkan adalah **Digital Input**, **Basic Counter**, **Debouncer**, **Digital Output**, **Not**, **Logic High '1'**, **Logic Low '0'**, dan **Clock**.

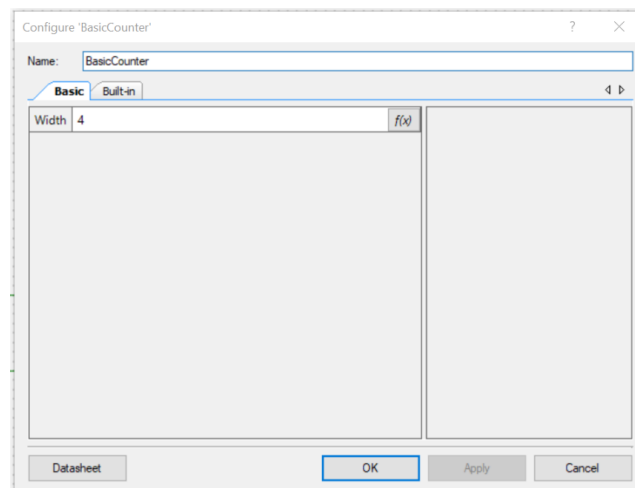


Gambar 1.1: Blok diagram *counter*

- Pengaturan blok Debouncer ditunjukkan oleh Gambar 1.2
- Pengaturan blok Basic Counter ditunjukkan oleh Gambar 1.3
- Atur konfigurasi pin seperti pada Gambar 1.4
- Lakukan *build* dan unduh program ke modul PSoC. Catat hasil pada Tabel Pengamatan



Gambar 1.2: Pengaturan blok Debouncer



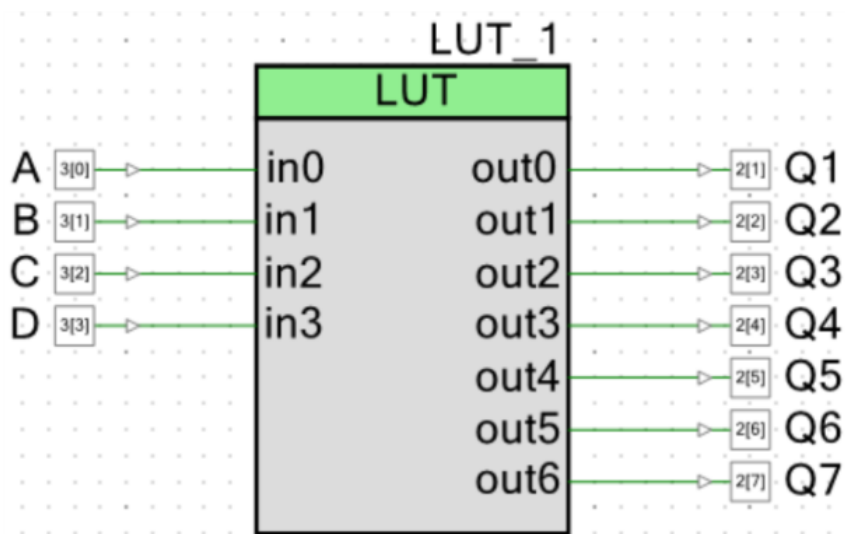
Gambar 1.3: Pengaturan blok Basic Counter

	Name	Port	Pin	Lock
<input type="checkbox"/>	Button	P0[7]	56	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Output0	P2[0]	62	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Output1	P2[1]	63	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Output2	P2[2]	64	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Output3	P2[3]	65	<input checked="" type="checkbox"/>

Gambar 1.4: Konfigurasi pin

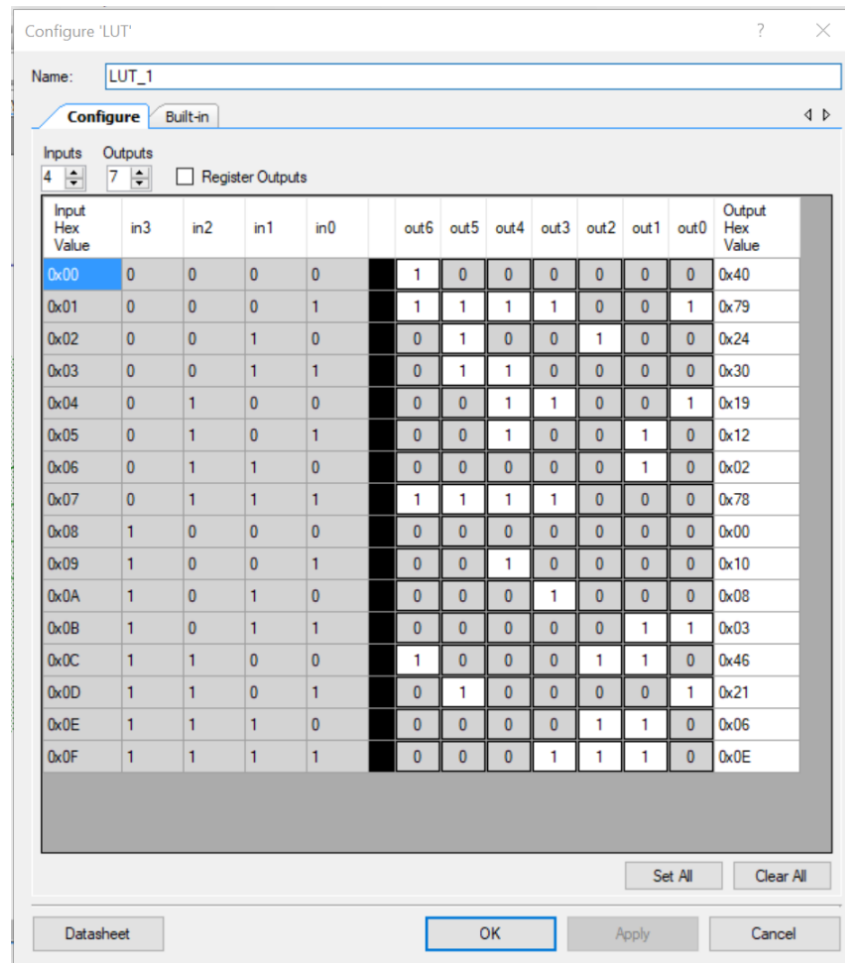
1.5.2 Seven segment

- Siapkan alat dan bahan yang dibutuhkan.
- Pastikan laptop telah menyala dan perangkat lunak PSoC Creator telah terbuka.
- Buat *Project* baru dengan nama *workspace* Modul_5_Seven_Segment.
- Buat rangkaian pada *worksheet* seperti pada Gambar 1.5. Blok-blok yang dibutuhkan adalah **Digital Input**, **Lookup Table**, dan **Digital Output**.



Gambar 1.5: Blok diagram *seven segment*

- Pengaturan blok **Lookup Table** ditunjukkan oleh Gambar 1.6.
- Atur konfigurasi pin seperti pada Gambar 1.7
- Lakukan *build* dan unduh program ke modul PSoC. Catat hasil pada Tabel Pengamatan.

Gambar 1.6: Pengaturan blok *Lookup Table common anode*

	Name	Port	Pin	Lock
<input checked="" type="checkbox"/>	A	P3[0]	29	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	B	P3[1]	30	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	C	P3[2]	31	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	D	P3[3]	32	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Q1	P2[1]	63	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Q2	P2[2]	64	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Q3	P2[3]	65	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Q4	P2[4]	66	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Q5	P2[5]	68	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Q6	P2[6]	1	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Q7	P2[7]	2	<input checked="" type="checkbox"/>

Gambar 1.7: Konfigurasi pin

TABEL PERCOBAAN

COUNTER

No.	Input ke- 5x	Output			
		QD	QC	QB	QA
1	1				
2	2				
3	3				
4	4				
5	5				
6	6				
7	7				
8	8				
9	9				
10	10				
11	11				
12	12				
13	13				
14	14				
15	15				

SEVEN SEGMENT

INPUT				OUTPUT
D	C	B	A	DISLAY 7 SEGMENT
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	

DAFTAR PUSTAKA

David Bucchlah, Wayne McLahan, “Applied Electronic Instrumentation And Measurement”, MacMilian Publishing Company, 1992.

Eggebrecht, Lewis C., Interfacing to The IBM PC, Howard W. Sams & Co., Indianapolis, 1987.

Hall, Douglas V., Microprocessor and Interfacing : Programming and Hardware, McGraw-Hill Book Company, Singapore, 1987.

Hodges D. , Jacson, Nasution S. “Analisa dan Desain Rangkaian Terpadu Digital”, Erlangga, Jakarta, 1987.

Ian Robertson Sinclair, Suryawan, “Panduan Belajar Elektronik Digital”, ElexMedia Komputindo, Jakarta, 1993.

K.F. Ibrahim, “Teknik Digital”, Andi Offset , Jakarta, 1996.

Sendra, Smith, Keneth C., “ Rangkaian Mikroelektronika”, Penerbit Erlangga, Jakarta, 1989.

Singh, Avtar & Walter A. Triebel, The 8088 Microprocessor : Programming, Interfacing, Software, Hardware and Applications, Prentice-Hall International Inc., New Jersey, 1987.

Sofyan H. Nasution, “Analisa dan Desain Rangkaian Terpadu Digital”, Penerbit Erlangga, Jakarta, 1987.

Sutrisno, “Rangkaian Digital dan Rancangan Logika”, Erlangga, Jakarta, 1990.

Tokheim. R., “Elektronika Digital”, Edisi Kedua, Erlangga, Jakarta, 1995.

Wijaya Widjanarka N., “Teknik Digital”, Erlangga, Jakarta, 2006.