

Rangkaian Sekuensial Sinkron

Kuliah#16 TKC205 - Sistem Digital

Eko Didik Widianto

Departemen Teknik Sistem Komputer, Universitas Diponegoro

11 Maret 2017

Finite State
Machine (FSM)

Metodologi Desain
Rangkaian FSM

Implementasi
dengan Flip-flop
Lain

Penyederhanaan
Rangkaian FSM

FSM Mealy

Ringkasan

Lisensi

Review Kuliah

Rangkaian
Sekuensial
Sinkron

@2017, Eko Didik
Widianto (didik@live.undip.ac.id)

- ▶ Kuliah sebelumnya telah dibahas rangkaian kombinasional dan sekuensial
- ▶ Bab ini membahas desain rangkaian sekuensial sinkron
 - ▶ Rangkaian ini bersifat sekuensial, yaitu keluaran rangkaian tergantung dari keadaan rangkaian sebelumnya dan membutuhkan elemen penyimpan berupa flip-flop
 - ▶ Rangkaian bersifat sinkron karena perilaku rangkaian dibangkitkan oleh transisi sumber detak yang sama, yaitu sinyal *Clk*
 - ▶ Dirancang menggunakan diagram FSM (Finite State Machine) atau mesin keadaan terbatas, sehingga disebut juga rangkaian FSM
 - ▶ Rangkaian FSM tersusun atas bagian kombinasional dan bagian sekuensial
 - ▶ Model FSM yang sering digunakan adalah model Moore dan Mealy.

Finite State
Machine (FSM)

Metodologi Desain
Rangkaian FSM

Implementasi
dengan Flip-flop
Lain

Penyederhanaan
Rangkaian FSM

FSM Mealy

Ringkasan

Lisensi

- ▶ FSM: diagram keadaan (*state diagram*) dan tabel keadaan (*state table*)
- ▶ desain FSM menggunakan model Moore
- ▶ implementasi FSM menggunakan DFF, TFF dan JKFF
- ▶ penyederhanaan rangkaian FSM dengan pemberian nilai keadaan
- ▶ desain FSM menggunakan model Mealy
- ▶ evaluasi rangkaian sekuensial sinkron

Kompetensi Dasar

- ▶ Setelah mempelajari bab ini, mahasiswa akan mampu:
 1. [C4] mengimplementasikan desain FSM menggunakan DFF
 2. [C4] mengimplementasikan desain FSM menggunakan TFF
 3. [C4] mengimplementasikan desain FSM menggunakan JKFF
 4. [C4] menyederhanakan rangkaian FSM dengan menerapkan aturan-aturan pemberian nilai keadaan yang dapat menghasilkan rangkaian yang lebih sederhana
 5. [C6] membuat rangkaian sekuensial sinkron dengan menerapkan FSM Moore dan Mealy menggunakan IC TTL
- ▶ Referensi:
 - ▶ Eko Didik Widiyanto, Sistem Digital: Analisis, Desain dan Implementasi, Penerbit Graha Ilmu, Cetakan 1, 2014 (**Bab 12**)
- ▶ Link

- ▶ Website: <http://didik.blog.undip.ac.id/2017/03/06/tkc205-sistem-digital-2016-genap/>

Buku Acuan/Referensi

Eko Didik Widiyanto, Sistem Digital: Analisis, Desain dan Implementasi, Edisi Pertama, Graha Ilmu, 2014
(Bab 12: Rangkaian Sekuensial Sinkron)

- ▶ Materi:
 - ▶ 12.1 Finite State Machine (FSM)
 - ▶ 12.2 Metodologi Desain Rangkaian FSM
 - ▶ 12.3 Implementasi Rangkaian dengan Flip-Flop
 - ▶ 12.4 Penyederhanaan Rangkaian FSM
 - ▶ 12.5 FSM Mealy
- ▶ Website:
 - ▶ <http://didik.blog.undip.ac.id/buku/sistem-digital/>



Rangkaian
Sekuensial
Sinkron

@2017, Eko Didik
Widiyanto (di-
dik@live.undip.ac.id)

Finite State
Machine (FSM)

Metodologi Desain
Rangkaian FSM

Implementasi
dengan Flip-flop
Lain

Penyederhanaan
Rangkaian FSM

FSM Mealy

Ringkasan

Lisensi

Bahasan

Finite State Machine (FSM)

Metodologi Desain Rangkaian FSM

Kebutuhan Desain

Diagram Keadaan

Tabel dan Variabel Keadaan

Peta Next-state dan Keluaran

Implementasi dan Analisis Rangkaian

Desain Pencacah

Implementasi dengan Flip-flop Lain

Implementasi dengan TFF

Implementasi dengan JKFF

Penyederhanaan Rangkaian FSM

Permasalahan Pemberian Nilai Keadaan

Petunjuk Pemberian Nilai Keadaan

One Hot Encoding

FSM Mealy

Ringkasan

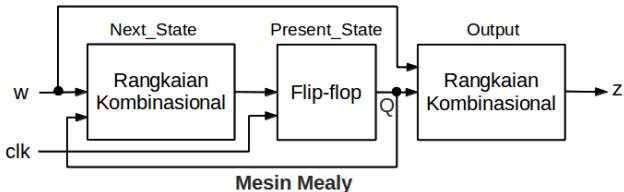
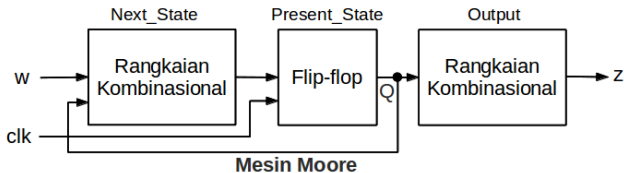
Lisensi

Rangkaian Sekuensial Sinkron

- ▶ **Rangkaian sekuensial sinkron:** rangkaian yang menggunakan sinyal clock untuk mengontrol operasi rangkaian
 - ▶ Transisi clock aktif menunjukkan transisi clock yang menyebabkan terjadinya perubahan state/keadaan. Transisi bisa positif atau negatif
 - ▶ Transisi positif: state dievaluasi saat transisi clock naik dari 0 ke 1
 - ▶ Transisi negatif: state dievaluasi saat transisi clock turun dari 1 ke 0
- ▶ Direalisasikan menggunakan rangkaian kombinasional dan setidaknya satu buah flip-flop
- ▶ Model rangkaian sekuensial sinkron:
 - ▶ **Moore:** keluaran rangkaian hanya tergantung pada state rangkaian saat ini
 - ▶ **Mealy:** keluaran rangkaian tergantung dari state rangkaian saat ini dan masukan primernya
- ▶ Rangkaian sekuensial disebut juga **finite state machine (FSM)**
 - ▶ **Mesin keadaan yang terbatas**

Finite State Machine (FSM)

- ▶ Mesin Moore menggunakan model Moore, sedangkan mesin Mealy menggunakan model Mealy



Mesin Moore dan Mealy

Jika masukan primer m bit dinyatakan w_m , keluaran $next_state$ n keadaan Y_n , keluaran $present_state$ n keadaan y_n dan keluaran $Output$ z , maka di mesin Moore:

- ▶ $next_state$: $Y_n = f(w_m, y_n)$
- ▶ $present_state$: untuk tiap transisi clk , $y_n = Y_n$
- ▶ $output$: $z = f(y_n)$

sedangkan di mesin Mealy:

- ▶ $next_state$: $Y_n = f(w_m, y_n)$
- ▶ $present_state$: untuk tiap transisi clk , $y_n = Y_n$
- ▶ $output$: $z = f(y_n, w_m)$

Langkah Desain Moore

1. Menganalisis kebutuhan desain;
2. Menggambar diagram keadaan (state diagram) dari kebutuhan: model Moore;
3. Menuangkan diagram keadaan ke dalam tabel keadaan (state table);
4. Menyatakan nilai variabel dari keadaan.
Aturan-aturan penentuan nilai variabel keadaan dilakukan untuk menghasilkan rangkaian yang lebih efisien;
5. Membuat peta Karnaugh untuk next-state sebagai fungsi dari present-state dan masukan utama. Dan nyatakan persamaan next-state;
6. Memuat peta Karnaugh untuk keluaran sebagai fungsi dari present-state. Dan nyatakan persamaan keluaran;
7. Menggambar rangkaian sekuensial sinkron menggunakan flip-flop dan gerbang logika;

Bahasan

Finite State Machine (FSM)

Metodologi Desain Rangkaian FSM

Kebutuhan Desain

Diagram Keadaan

Tabel dan Variabel Keadaan

Peta Next-state dan Keluaran

Implementasi dan Analisis Rangkaian

Desain Pencacah

Implementasi dengan Flip-flop Lain

Implementasi dengan TFF

Implementasi dengan JKFF

Penyederhanaan Rangkaian FSM

Permasalahan Pemberian Nilai Keadaan

Petunjuk Pemberian Nilai Keadaan

One Hot Encoding

FSM Mealy

Ringkasan

Rangkaian
Sekuensial
Sinkron

@2017,Eko Didik
Widianto (di-
dik@live.undip.ac.id)

Finite State
Machine (FSM)

Metodologi Desain
Rangkaian FSM

Kebutuhan Desain

Diagram Keadaan

Tabel dan Variabel Keadaan

Peta Next-state dan
Keluaran

Implementasi dan Analisis
Rangkaian

Desain Pencacah

Implementasi
dengan Flip-flop
Lain

Penyederhanaan
Rangkaian FSM

FSM Mealy

Ringkasan

Lisensi

Kebutuhan Rangkaian Sekuensial

Rangkaian
Sekuensial
Sinkron

@2017, Eko Didik
Widianto (di-
dik@live.undip.ac.id)

- ▶ Desain rangkaian sekuensial sinkron yang memenuhi spesifikasi berikut:
 - ▶ Rangkaian mempunyai satu masukan, w , dan satu keluaran, z
 - ▶ Semua perubahan dalam rangkaian terjadi saat transisi naik dari sinyal clock
 - ▶ Keluaran $z=1$ jika masukan $w=1$ secara berurutan selama 2 clock
- ▶ Dari spesifikasi terlihat bahwa keluaran z tidak hanya tergantung dari nilai w semata

Finite State
Machine (FSM)

Metodologi Desain
Rangkaian FSM

Kebutuhan Desain

Diagram Keadaan

Tabel dan Variabel Keadaan

Peta Next-state dan
Keluaran

Implementasi dan Analisis
Rangkaian

Desain Pencacah

Implementasi
dengan Flip-flop
Lain

Penyederhanaan
Rangkaian FSM

FSM Mealy

Ringkasan

Lisensi

- ▶ Urutan sinyal masukan dan keluaran berikut mendeskripsikan rangkaian yang diinginkan

Siklus clock	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
w	0	1	0	1	1	1	0	1	1	0	1
z	0	0	0	0	0	1	1	0	0	1	0

- ▶ Siklus detak terjadi setiap transisi naik Clk
 - ▶ Nilai $z=1$ di t_5 , t_6 dan t_9 karena masukan w di 2 siklus sebelumnya bernilai 1 atau terjadi urutan masukan $w = 1$ selama 2 detak Clk

Bahasan

Finite State Machine (FSM)

Metodologi Desain Rangkaian FSM

Kebutuhan Desain

Diagram Keadaan

Tabel dan Variabel Keadaan

Peta Next-state dan Keluaran

Implementasi dan Analisis Rangkaian

Desain Pencacah

Implementasi dengan Flip-flop Lain

Implementasi dengan TFF

Implementasi dengan JKFF

Penyederhanaan Rangkaian FSM

Permasalahan Pemberian Nilai Keadaan

Petunjuk Pemberian Nilai Keadaan

One Hot Encoding

FSM Mealy

Ringkasan

Rangkaian
Sekuensial
Sinkron

@2017,Eko Didik
Widianto (di-
dik@live.undip.ac.id)

Finite State
Machine (FSM)

Metodologi Desain
Rangkaian FSM

Kebutuhan Desain

Diagram Keadaan

Tabel dan Variabel Keadaan

Peta Next-state dan
Keluaran

Implementasi dan Analisis
Rangkaian

Desain Pencacah

Implementasi
dengan Flip-flop
Lain

Penyederhanaan
Rangkaian FSM

FSM Mealy

Ringkasan

Lisensi

Diagram Keadaan

- ▶ Langkah pertama untuk mendesain FSM adalah dengan menentukan **berapa keadaan** (*state*) yang dibutuhkan dan **transisi apa yang mungkin** dari satu keadaan ke keadaan lain
 - ▶ Tanpa prosedur preset untuk tiap keadaan (bersifat sekuensial)
 - ▶ Desainer harus berpikir tentang rangkaian untuk memenuhi FSM ini
- ▶ Desain dimulai dengan mendefinisikan keadaan reset, yaitu keadaan saat sumber daya diberikan ke rangkaian atau sinyal reset diterima

Diagram Keadaan (1)

- ▶ Asumsi, keadaan mulai adalah A, yaitu keadaan saat reset
- ▶ Selama masukan $w=0$, rangkaian tidak melakukan apapun dan nilai $z=0$

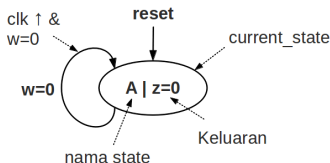


Diagram Keadaan (2)

- ▶ Saat $w=1$, rangkaian akan **mengalami transisi** ke keadaan baru (**B**)
 - ▶ Transisi ini terjadi di transisi naik sinyal clock berikutnya
 - ▶ Di keadaan B ini, nilai keluaran z masih 0 ($z=0$)

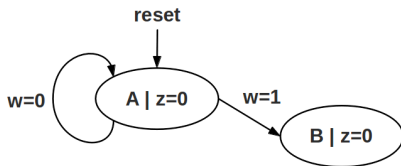


Diagram Keadaan (3)

- ▶ Saat rangkaian berada di keadaan B dan $w=1$, rangkaian akan mengalami transisi ke keadaan baru (C)
 - ▶ Transisi ini terjadi di transisi naik sinyal clock berikutnya
 - ▶ Di keadaan C ini, nilai keluaran z menjadi 1 ($z=1$) karena terjadi deretan w yang bernilai 11
- ▶ Jika $w=0$? maka keadaan akan kembali ke A dan deretan harus mulai dari keadaan A

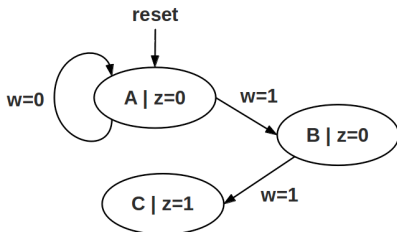
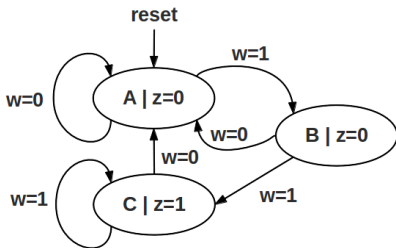


Diagram Keadaan (4)

- ▶ Diagram keadaan lengkap (Model Moore)
 - ▶ Terdiri atas 3 keadaan A, B dan C untuk semua valuasi nilai masukan w



Bahasan

Finite State Machine (FSM)

Metodologi Desain Rangkaian FSM

Kebutuhan Desain

Diagram Keadaan

Tabel dan Variabel Keadaan

Peta Next-state dan Keluaran

Implementasi dan Analisis Rangkaian

Desain Pencacah

Implementasi dengan Flip-flop Lain

Implementasi dengan TFF

Implementasi dengan JKFF

Penyederhanaan Rangkaian FSM

Permasalahan Pemberian Nilai Keadaan

Petunjuk Pemberian Nilai Keadaan

One Hot Encoding

FSM Mealy

Ringkasan

Rangkaian
Sekuensial
Sinkron

@2017,Eko Didik
Widianto (di-
dik@live.undip.ac.id)

Finite State
Machine (FSM)

Metodologi Desain
Rangkaian FSM

Kebutuhan Desain

Diagram Keadaan

Tabel dan Variabel Keadaan

Peta Next-state dan

Keluaran

Implementasi dan Analisis

Rangkaian

Desain Pencacah

Implementasi
dengan Flip-flop
Lain

Penyederhanaan
Rangkaian FSM

FSM Mealy

Ringkasan

Lisensi

Tabel Keadaan (*State Table*)

- ▶ Diagram keadaan menjabarkan fungsionalitas rangkaian (bukan implementasinya)
 - ▶ Translasi ke bentuk tabular menjadi tabel keadaan (*state table*)
- ▶ Tabel keadaan berisi:
 - ▶ Semua transisi dari tiap **present_state** ke **next_state** untuk semua valuasi dari nilai masukan (**w**)
 - ▶ Keluaran, **z**, ditentukan oleh **present_state** (mesin Moore)

Present state	Next_state		Output z
	w=0	w=1	
A	A	B	0
B	A	C	0
C	A	C	1

Pemberian Nilai Keadaan (*State Assignment*)

- ▶ Keadaan didefinisikan sebagai variabel, misalkan keadaan **A**, **B**, dan **C**
- ▶ Tiap keadaan direpresentasikan dengan valuasi (pemberian nilai) dari **variabel keadaan** tersebut
- ▶ Tiap variabel state diimplementasikan dengan **sebuah flip-flop**
- ▶ Karena terdapat 3 state yang perlu direalisasikan, maka akan diperlukan dua buah variabel state
 - ▶ Menggunakan y_2y_1 untuk *present state* (sebagai variabel *present_state*)
 - ▶ Menggunakan Y_2Y_1 untuk *next state* (sebagai variabel *next_state*)

Tabel Keadaan Bernilai (*State-assigned Table*)

- ▶ Valuasi variabel keadaan membentuk tabel keadaan bernilai
 - ▶ Keadaan A=00, B=01 dan C=10

State	Valuasi	Present state	Next state $Y_2 Y_1$		Output
		$y_2 y_1$	w=0	w=1	
A	00	00	00	01	0
B	01	01	00	10	0
C	10	10	00	10	1
		11	dd	dd	d

- ▶ Keadaan D, $y_2 y_1 = 11$ ditambahkan dalam tabel
 - ▶ Tidak digunakan, namun diperlukan untuk kelengkapan

Bahasan

Finite State Machine (FSM)

Metodologi Desain Rangkaian FSM

Kebutuhan Desain

Diagram Keadaan

Tabel dan Variabel Keadaan

Peta Next-state dan Keluaran

Implementasi dan Analisis Rangkaian

Desain Pencacah

Implementasi dengan Flip-flop Lain

Implementasi dengan TFF

Implementasi dengan JKFF

Penyederhanaan Rangkaian FSM

Permasalahan Pemberian Nilai Keadaan

Petunjuk Pemberian Nilai Keadaan

One Hot Encoding

FSM Mealy

Ringkasan

Rangkaian
Sekuensial
Sinkron

@2017,Eko Didik
Widianto (di-
dik@live.undip.ac.id)

Finite State
Machine (FSM)

Metodologi Desain
Rangkaian FSM

Kebutuhan Desain

Diagram Keadaan

Tabel dan Variabel Keadaan

**Peta Next-state dan
Keluaran**

Implementasi dan Analisis
Rangkaian

Desain Pencacah

Implementasi
dengan Flip-flop
Lain

Penyederhanaan
Rangkaian FSM

FSM Mealy

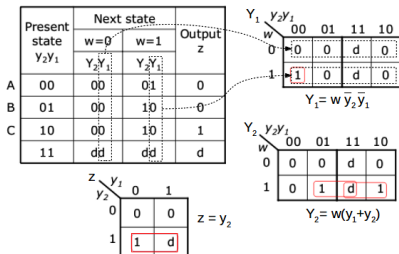
Ringkasan

Lisensi

Peta Next-state dan Keluaran

- ▶ Peta Karnaugh disusun dari tabel nilai keadaan untuk:
 - ▶ Rangkaian luaran (z)
 - ▶ Rangkaian masukan ke flip-flop (next-state)
- ▶ Rangkaian luaran dan rangkaian masukan flip-flop adalah rangkaian kombinasional
- ▶ Mengkonstruksi peta next-state tergantung dari tipe flip-flop (D, T, JK) yang akan digunakan untuk implementasi
 - ▶ Flip-flop D dapat digunakan secara langsung
 - ▶ Peta next-state **disusun secara langsung** dari tabel keadaan bernilai karena $Q(t + 1) = Q^+ = D$

Peta Next_state dan Keluaran



- Di rangkaian flip-flop D, nilai $Y_2Y_1 = D_2D_1$ dengan D_x adalah masukan flip-flop data x

$$Y_1 = w\bar{y}_2\bar{y}_1$$

$$Y_2 = wy_1 + wy_2 = w(y_1 + y_2)$$

$$z = y_2$$

Bahasan

Finite State Machine (FSM)

Metodologi Desain Rangkaian FSM

Kebutuhan Desain

Diagram Keadaan

Tabel dan Variabel Keadaan

Peta Next-state dan Keluaran

Implementasi dan Analisis Rangkaian

Desain Pencacah

Implementasi dengan Flip-flop Lain

Implementasi dengan TFF

Implementasi dengan JKFF

Penyederhanaan Rangkaian FSM

Permasalahan Pemberian Nilai Keadaan

Petunjuk Pemberian Nilai Keadaan

One Hot Encoding

FSM Mealy

Ringkasan

Rangkaian
Sekuensial
Sinkron

@2017,Eko Didik
Widianto (di-
dik@live.undip.ac.id)

Finite State
Machine (FSM)

Metodologi Desain
Rangkaian FSM

Kebutuhan Desain

Diagram Keadaan

Tabel dan Variabel Keadaan

Peta Next-state dan
Keluaran

Implementasi dan Analisis
Rangkaian

Desain Pencacah

Implementasi
dengan Flip-flop
Lain

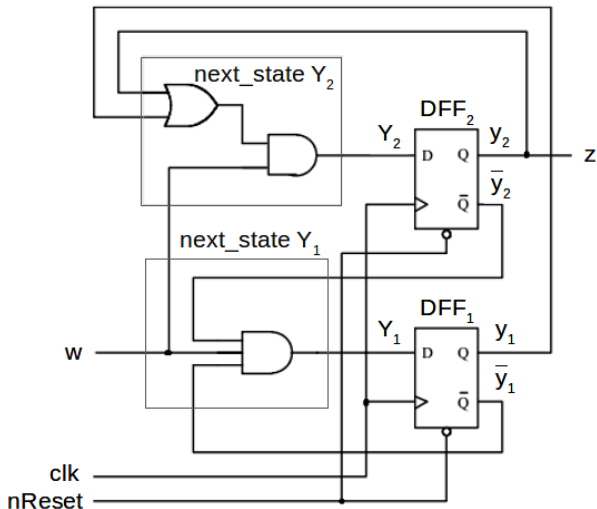
Penyederhanaan
Rangkaian FSM

FSM Mealy

Ringkasan

Lisensi

Diagram Rangkaian



Rangkaian
Sekuensial
Sinkron

@2017, Eko Didik
Widiyanto (didik@live.undip.ac.id)

Finite State
Machine (FSM)

Metodologi Desain
Rangkaian FSM

Kebutuhan Desain
Diagram Keadaan
Tabel dan Variabel Keadaan
Peta Next-state dan
Keluaran

Implementasi dan Analisis
Rangkaian
Desain Pencacah

Implementasi
dengan Flip-flop
Lain

Penyederhanaan
Rangkaian FSM

FSM Mealy

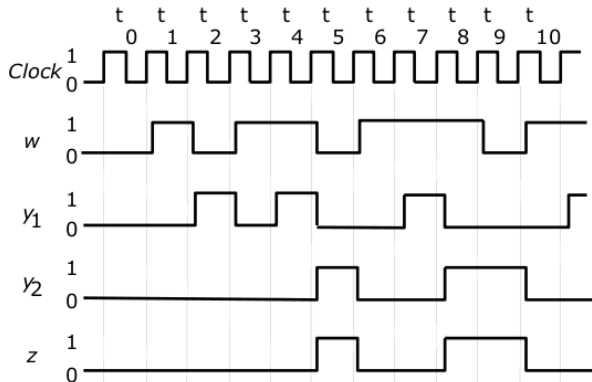
Ringkasan

Lisensi

Diagram Pewaktuan

Rangkaian
Sekuensial
Sinkron

@2017, Eko Didik
Widianto (didik@live.undip.ac.id)



Finite State
Machine (FSM)

Metodologi Desain
Rangkaian FSM

Kebutuhan Desain
Diagram Keadaan
Tabel dan Variabel Keadaan
Peta Next-state dan
Keluaran
Implementasi dan Analisis
Rangkaian
Desain Pencacah

Implementasi
dengan Flip-flop
Lain

Penyederhanaan
Rangkaian FSM

FSM Mealy

Ringkasan

Lisensi

Bahasan

Finite State Machine (FSM)

Metodologi Desain Rangkaian FSM

Kebutuhan Desain

Diagram Keadaan

Tabel dan Variabel Keadaan

Peta Next-state dan Keluaran

Implementasi dan Analisis Rangkaian

Desain Pencacah

Implementasi dengan Flip-flop Lain

Implementasi dengan TFF

Implementasi dengan JKFF

Penyederhanaan Rangkaian FSM

Permasalahan Pemberian Nilai Keadaan

Petunjuk Pemberian Nilai Keadaan

One Hot Encoding

FSM Mealy

Ringkasan

Rangkaian
Sekuensial
Sinkron

@2017,Eko Didik
Widianto (di-
dik@live.undip.ac.id)

Finite State
Machine (FSM)

Metodologi Desain
Rangkaian FSM

Kebutuhan Desain

Diagram Keadaan

Tabel dan Variabel Keadaan

Peta Next-state dan
Keluaran

Implementasi dan Analisis
Rangkaian

Desain Pencacah

Implementasi
dengan Flip-flop
Lain

Penyederhanaan
Rangkaian FSM

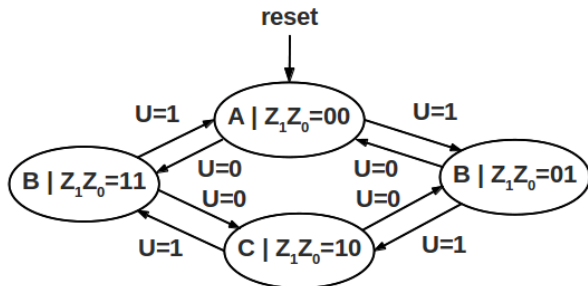
FSM Mealy

Ringkasan

Lisensi

- ▶ Desain pencacah 2-bit untuk mencacah dengan ketentuan berikut:
 - ▶ Urutan 0,1,2,3,0,... (*up counter*) jika sinyal kontrol $U=1$ atau
 - ▶ Urutan 0,3,2,1,0,... (*down counter*) jika sinyal kontrol $U=0$
- ▶ Desain ini membuat pencacah naik/turun 2-bit
 - ▶ Masukan U mengontrol arah pencacahan (naik/turun)
 - ▶ Masukan Reset mereset pencacah ke NOL
 - ▶ Dua keluaran ($Z_1 Z_0$) menunjukkan nilai keluaran (0-3)
 - ▶ Pencacah menghitung saat transisi positif sinyal clock
- ▶ Tujuan: mendesain rangkaian pencacah sebagai mesin sekuensial sinkron menggunakan
 - ▶ Flip-flop D, T, dan JK

Diagram Keadaan Pencacah



Tabel Keadaan Pencacah

Rangkaian
Sekuensial
Sinkron

@2017,Eko Didik
Widianto (di-
dik@live.undip.ac.id)

Present state	Next state		Output $Z_1 Z_0$
	U=0	U=1	
A	D	B	00
B	A	C	01
C	B	D	10
D	C	A	11

Finite State
Machine (FSM)

Metodologi Desain
Rangkaian FSM

Kebutuhan Desain

Diagram Keadaan

Tabel dan Variabel Keadaan

Peta Next-state dan
Keluaran

Implementasi dan Analisis
Rangkaian

Desain Pencacah

Implementasi
dengan Flip-flop
Lain

Penyederhanaan
Rangkaian FSM

FSM Mealy

Ringkasan

Lisensi

Tabel Keadaan Bernilai

- ▶ Misalnya:
 - ▶ Memilih pemberian nilai keadaan A=00, B=01, C=10 dan D=11
 - ▶ Keluaran Z_1Z_0 menjadi output dari flip-flop secara langsung (nilai variabel *present_state*)

Present state y_2y_1	Next state Y_2Y_1		Output Z_1Z_0
	U=0	U=1	
00	11	01	00
01	00	10	01
10	01	11	10
11	10	00	11

Implementasi dengan Flip-Flip D

- ▶ Untuk implementasi FSM dengan flip-flop D, sinyal **next-state** dalam tabel pemberian nilai keadaan berkaitan secara langsung dengan sinyal yang harus diaplikasikan ke **masukan D**
 - ▶ atau $D_x = Y_x$
- ▶ Kemudian peta Karnaugh untuk masukan D dapat diturunkan secara langsung dari tabel keadaan bernilai
- ▶ Mekanisme ini tidak berlaku untuk tipe flip-flop lain (T, JK)

Tabel Keadaan Bernilai dan Peta Next state

	Present state y_2y_1	Next state		Output Z_1Z_0
		U=0	U=1	
		Y_2Y_1	Y_2Y_1	
A	00	11	01	00
B	01	00	10	01
C	10	01	11	10
D	11	10	00	11

$$Z_1=y_2 \quad Z_0=y_1$$

		y_2y_1			
		00	01	11	10
u	0	1	0	0	1
	1	1	0	0	1

$$Y_1=y_1'$$

		y_2y_1			
		00	01	11	10
u	0	1	0	1	0
	1	0	1	0	1

$$Y_2=(y_2 \oplus y_1 \oplus u)'$$

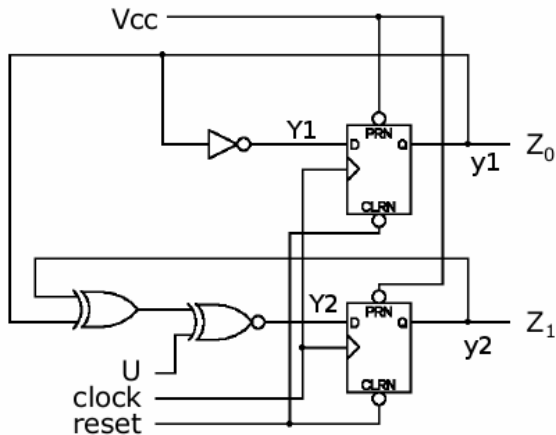
$$Y_1 = \bar{y}_1$$

$$Y_2 = \overline{y_1 \oplus y_2 \oplus u}$$

$$Z_0 = y_1$$

$$Z_1 = y_2$$

Diagram Rangkaian (Flip-flop D)



Rangkaian
Sekuensial
Sinkron

@2017, Eko Didik
Widiyanto (didik@live.undip.ac.id)

Finite State
Machine (FSM)

Metodologi Desain
Rangkaian FSM

Kebutuhan Desain
Diagram Keadaan
Tabel dan Variabel Keadaan
Peta Next-state dan
Keluaran
Implementasi dan Analisis
Rangkaian
Desain Pencacah

Implementasi
dengan Flip-flop
Lain

Penyederhanaan
Rangkaian FSM

FSM Mealy

Ringkasan

Lisensi

Implementasi dengan Flip-flop Lain

- ▶ Implementasi FSM menggunakan DFF dapat dilakukan dengan membuat secara langsung K-map untuk fungsi *next_state* dari tabel keadaan bernilai
 - ▶ Keluaran fungsi *next_state* ini langsung diimplementasikan ke masukan DFF sehingga $D_x = Y_x$
 - ▶ Tidak berlaku untuk TFF dan JK
- ▶ Untuk flip-flop T- dan JK-, langkah yang perlu dilakukan adalah menurunkan input yang diinginkan ke dalam flip-flop
 - ▶ Dimulai dengan mengkonstruksi **tabel transisi** untuk flip-flop yang akan digunakan
 - ▶ Tabel ini menyederhanakan daftar masukan yang diinginkan untuk satu perubahan keadaan
 - ▶ Tabel transisi ini digunakan dengan tabel keadaan bernilai untuk menyusun sebuah **tabel eksitasi**
 - ▶ Tabel eksitasi menentukan masukan flip-flop yang diperlukan yang harus dieksitasi untuk menyebabkan transisi ke masukan berikutnya

Tabel Transisi

- ▶ Tabel transisi mendaftarkan masukan flip-flop yang mempengaruhi perubahan tertentu
 - ▶ Diturunkan dari **tabel karakteristik** flip-flop yang digunakan
 - ▶ Tabel ini menunjukkan nilai masukan untuk tiap kemungkinan perubahan keadaan dari Q ke Q^+

J	K	Q	Q ⁺	Q	Q ⁺	J	K
0	0	0	0	0	0	0	D
0	0	1	1	0	1	1	D
0	1	0	0	1	0	D	1
0	1	1	0	1	1	D	0
1	0	0	1	JK transition table			
1	0	1	1				
1	1	0	1				
1	1	1	0				

T	Q	Q ⁺	Q	Q ⁺	T
0	0	0	0	0	0
0	1	1	0	1	1
1	0	1	1	0	1
1	1	0	1	1	0
			T transition table		

Bahasan

Finite State Machine (FSM)

Metodologi Desain Rangkaian FSM

Kebutuhan Desain

Diagram Keadaan

Tabel dan Variabel Keadaan

Peta Next-state dan Keluaran

Implementasi dan Analisis Rangkaian

Desain Pencacah

Implementasi dengan Flip-flop Lain

Implementasi dengan TFF

Implementasi dengan JKFF

Penyederhanaan Rangkaian FSM

Permasalahan Pemberian Nilai Keadaan

Petunjuk Pemberian Nilai Keadaan

One Hot Encoding

FSM Mealy

Ringkasan

Rangkaian
Sekuensial
Sinkron

@2017, Eko Didik
Widianto (didik@live.undip.ac.id)

Finite State
Machine (FSM)

Metodologi Desain
Rangkaian FSM

Implementasi
dengan Flip-flop
Lain

Implementasi dengan TFF
Implementasi dengan JKFF

Penyederhanaan
Rangkaian FSM

FSM Mealy

Ringkasan

Lisensi

Implementasi dengan Flip-flop T

- ▶ Menggunakan daftar dari tabel transisi untuk menurunkan masukan flip-flop berdasarkan tabel keadaan bernilai (mis. desain pencacah 2 bit)
 - ▶ Membentuk **tabel eksitasi**

Q	Q+	T	Present state y_2y_1	Masukan TFF				Output Z_1Z_0
				$U = 0$		$U = 1$		
				Y_2Y_1	T_2T_1	Y_2Y_1	T_2T_1	
0	0	0	00	11	11	01	01	00
0	1	1	01	00	01	10	11	01
1	0	1	10	01	11	11	01	10
1	1	0	11	10	01	00	11	11

- ▶ Nilai T_2T_1 diperoleh dengan mencari nilai T dari masukan $y_x Y_x$ dalam tabel transisi
 - ▶ jika $y_2y_1 = 00$ dan $Y_2Y_1 = 11$, maka akan menghasilkan $T_2T_1 = 11$ karena $y_2Y_2 = 01$ dan $y_1Y_1 = 01$ menghasilkan

Tabel Eksitasi dan Peta Karnaugh

- ▶ Mencari masukan T_1 untuk flip-flop #1 dan T_2 untuk flip-flop #2
- ▶ Kolom **next_state** diabaikan sehingga diperoleh tabel eksitasi

Present state Y_2Y_1	Flip-flop inputs		Output Z_1Z_0
	U=0	U=1	
	T_2T_1	T_2T_1	
00	11	01	00
01	01	11	01
10	11	01	10
11	01	11	11

$$Z_1=y_2 \quad Z_0=y_1$$

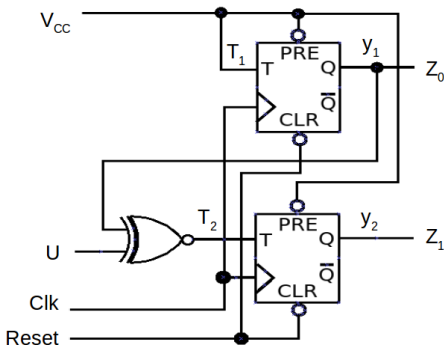
y_2y_1	00	01	11	10
0	1	1	1	1
1	1	1	1	1

$$T_1=1$$

y_2y_1	00	01	11	10
0	1	0	0	1
1	0	1	1	0

$$T_2=y_1u+y_1'u'=(y_1\oplus u)'$$

Diagram Rangkaian (TFF)



► Latihan

- Desain rangkaian sekuensial sinkron untuk pendeteksi urutan $1 \rightarrow 1 \rightarrow 0$ menggunakan TFF. Gambarkan skematik rangkaiannya menggunakan IC TTL. Analisis rangkaian sehingga jelas perilakunya sebagai pendeteksi urutan $1 \rightarrow 1 \rightarrow 0$

Bahasan

Finite State Machine (FSM)

Metodologi Desain Rangkaian FSM

Kebutuhan Desain

Diagram Keadaan

Tabel dan Variabel Keadaan

Peta Next-state dan Keluaran

Implementasi dan Analisis Rangkaian

Desain Pencacah

Implementasi dengan Flip-flop Lain

Implementasi dengan TFF

Implementasi dengan JKFF

Penyederhanaan Rangkaian FSM

Permasalahan Pemberian Nilai Keadaan

Petunjuk Pemberian Nilai Keadaan

One Hot Encoding

FSM Mealy

Ringkasan

Rangkaian
Sekuensial
Sinkron

@2017, Eko Didik
Widianto (didik@live.undip.ac.id)

Finite State
Machine (FSM)

Metodologi Desain
Rangkaian FSM

Implementasi
dengan Flip-flop
Lain

Implementasi dengan TFF
Implementasi dengan JKFF

Penyederhanaan
Rangkaian FSM

FSM Mealy

Ringkasan

Lisensi

Implementasi flip-flop JK

- ▶ Menggunakan daftar dari tabel transisi untuk menurunkan masukan flip-flop berdasarkan tabel keadaan bernilai
 - ▶ Harus dilakukan untuk tiap masukan (J dan K) di tiap flip-flop

Present state Y_2Y_1	Next state		Output Z_1Z_0
	U=0	U=1	
	Y_2Y_1	Y_2Y_1	
00	11	01	00
01	00	10	01
10	01	11	10
11	10	00	11

Q	Q ⁺	J	K
0	0	0	D
0	1	1	D
1	0	D	1
1	1	D	0

Tabel Transisi
JK

Implementasi flip-flop JK (2)

- ▶ Membuat tabel eksitasi
 - ▶ Tiap kolom next_state diuraikan menjadi 2 kolom masukan JK, yaitu J_2K_2 untuk flip-flop #2 dan J_1K_1 untuk flip-flop #1

Q	Q ⁺	J	K
0	0	0	D
0	1	1	D
1	0	D	1
1	1	D	0

Tabel Transisi JK

Present state Y_2Y_1	Flip-flop inputs						Output Z_1Z_0
	U=0			U=1			
	Y_2Y_1	J_2K_2	J_1K_1	Y_2Y_1	J_2K_2	J_1K_1	
00	11	1D	1D	01	0D	1D	00
01	00	0D	D1	10	1D	D1	01
10	01	D1	1D	11	D0	1D	10
11	10	D0	D1	00	D1	D1	11

Tabel Eksitasi dan Peta Karnaugh

- Mencari masukan J dan K untuk flip-flop #1

Present state Y_2Y_1	Flip-flop inputs						Output Z_1Z_0
	U=0			U=1			
	Y_2Y_1	J_2K_2	J_1K_1	Y_2Y_1	J_2K_2	J_1K_1	
00	11	1D	1D	01	0D	1D	00
01	00	0D	D1	10	1D	D1	01
10	01	D1	1D	11	D0	1D	10
11	10	D0	D1	00	D1	D1	11

Karnaugh map for J_1 :

u	y_2y_1	00	01	11	10
0	0	1	D	D	1
1	1	1	D	D	1

$$J_1 = 1$$

Karnaugh map for K_1 :

u	y_2y_1	00	01	11	10
0	0	D	1	1	D
1	1	D	1	1	D

$$K_1 = 1$$

Tabel Eksitasi dan Peta Karnaugh (2)

- Mencari masukan J dan K untuk flip-flop #2

Present state Y_2Y_1	Flip-flop inputs						Output Z_1Z_0
	U=0			U=1			
	Y_2Y_1	J_2K_2	J_1K_1	Y_2Y_1	J_2K_2	J_1K_1	
00	11	1D	1D	01	0D	1D	00
01	00	0D	D1	10	1D	D1	01
10	01	D1	1D	11	D0	1D	10
11	10	D0	D1	00	D1	D1	11

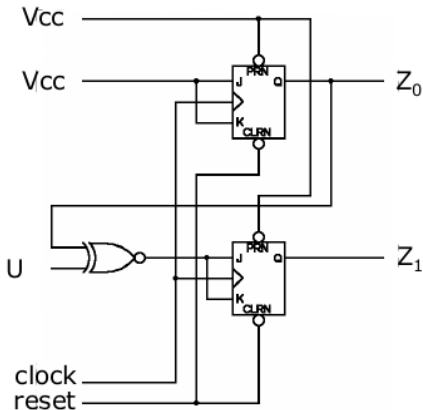
u	y_2y_1	00	01	11	10
0		1	0	D	D
1		0	1	D	D

$$J_2 = (y_1 \oplus u)'$$

u	y_2y_1	00	01	11	10
0		D	D	0	1
1		D	D	1	0

$$K_2 = (y_1 \oplus u)'$$

Diagram Rangkaian (Flip-flop JK)



- ▶ Desain rangkaian sekuensial sinkron untuk pendeteksi urutan 1 → 1 → 0 menggunakan JKFF. Gambarkan skematik rangkaiannya menggunakan IC TTL. Analisis rangkaian sehingga jelas perilakunya sebagai pendeteksi urutan 1 → 1 → 0

Bahasan

Finite State Machine (FSM)

Metodologi Desain Rangkaian FSM

Kebutuhan Desain

Diagram Keadaan

Tabel dan Variabel Keadaan

Peta Next-state dan Keluaran

Implementasi dan Analisis Rangkaian

Desain Pencacah

Implementasi dengan Flip-flop Lain

Implementasi dengan TFF

Implementasi dengan JKFF

Penyederhanaan Rangkaian FSM

Permasalahan Pemberian Nilai Keadaan

Petunjuk Pemberian Nilai Keadaan

One Hot Encoding

FSM Mealy

Ringkasan

Rangkaian
Sekuensial
Sinkron

@2017, Eko Didik
Widianto (di-
dik@live.undip.ac.id)

Finite State
Machine (FSM)

Metodologi Desain
Rangkaian FSM

Implementasi
dengan Flip-flop
Lain

Penyederhanaan
Rangkaian FSM

Permasalahan Pemberian
Nilai Keadaan

Petunjuk Pemberian Nilai
Keadaan

One Hot Encoding

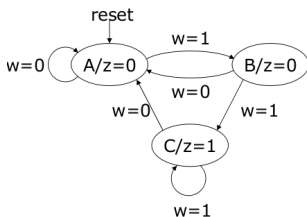
FSM Mealy

Ringkasan

Lisensi

Permasalahan Pemberian Nilai Keadaan

- ▶ Di contoh sebelumnya, pemberian nilai keadaan yang dilakukan masih sederhana dan secara langsung
 - ▶ Keadaan A diberi nilai 00, B dengan 01, C dengan 10 dan seterusnya
 - ▶ Bagaimana melakukan pemberian nilai alternatif untuk keadaan yang dapat menghasilkan solusi rangkaian yang lebih sederhana?



	Present state Y_2Y_1	Next state		Output z
		$w=0$	$w=1$	
		Y_2Y_1	Y_2Y_1	
A	00	00	01	0
B	01	00	10	0
C	10	00	10	1
	11	dd	dd	d

Pemberian Nilai Keadaan Alternatif

- ▶ Dengan mengubah nilai keadaan

Keadaan	Nilai semula	Nilai Alternatif
A	00	00
B	01	01
C	10	11
Tidak digunakan	11	10

- ▶ Tabel nilai keadaannya menjadi:

Present state	Next state		Output z
	w=0	w=1	
A	A	B	0
B	A	C	0
C	A	C	1

	Present state Y_2Y_1	Next state		Output z
		w=0 Y_2Y_1	w=1 Y_2Y_1	
A	00	00	01	0
B	01	00	11	0
C	11	00	11	1
	10	dd	dd	d

K-map untuk fungsi next_state dan Keluaran

	Present state Y_2Y_1	Next state		Output z
		$w=0$	$w=1$	
		Y_2Y_1	Y_2Y_1	
A	00	00	01	0
B	01	00	11	0
C	11	00	11	1
	10	dd	dd	d

Y_2

	$y_2 y_1$ 00	01	11	10
w 0	0	0	0	d
1	0	1	1	d

$w.y_1$

Y_1

	$y_2 y_1$ 00	01	11	10
w 0	0	0	0	d
1	1	1	1	d

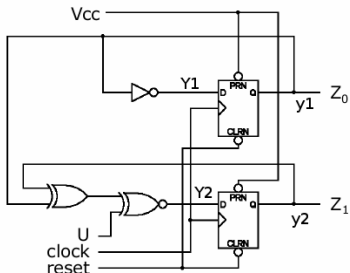
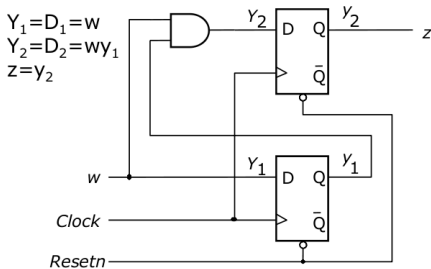
w

z

	y_1 0	1
y_2 0	0	0
1	d	1

y_2

Implementasi Rangkaian yang Disederhanakan



Bandingkan dengan:

Bahasan

Finite State Machine (FSM)

Metodologi Desain Rangkaian FSM

Kebutuhan Desain

Diagram Keadaan

Tabel dan Variabel Keadaan

Peta Next-state dan Keluaran

Implementasi dan Analisis Rangkaian

Desain Pencacah

Implementasi dengan Flip-flop Lain

Implementasi dengan TFF

Implementasi dengan JKFF

Penyederhanaan Rangkaian FSM

Permasalahan Pemberian Nilai Keadaan

Petunjuk Pemberian Nilai Keadaan

One Hot Encoding

FSM Mealy

Ringkasan

Rangkaian
Sekuensial
Sinkron

@2017, Eko Didik
Widianto (di-
dik@live.undip.ac.id)

Finite State
Machine (FSM)

Metodologi Desain
Rangkaian FSM

Implementasi
dengan Flip-flop
Lain

Penyederhanaan
Rangkaian FSM

Permasalahan Pemberian
Nilai Keadaan

Petunjuk Pemberian Nilai
Keadaan

One Hot Encoding

FSM Mealy

Ringkasan

Lisensi

Permasalahan Pemberian Nilai Keadaan

Rangkaian
Sekuensial
Sinkron

@2017, Eko Didik
Widianto (di-
dik@live.undip.ac.id)

- ▶ Secara umum, untuk rangkaian yang lebih besar dan kompleks, pemberian nilai keadaan yang berbeda dapat sangat berpengaruh ke biaya implementasi rangkaian akhirnya
 - ▶ Pemberian nilai keadaan **merupakan langkah kritis**
 - ▶ Bahkan seringkali tidak mungkin (tidak dapat dilakukan) untuk menemukan pemberian nilai variabel terbaik untuk rangkaian yang besar karena jumlah keadaan yang tersedia banyak
 - ▶ Perangkat CAD digunakan untuk membuat pemberian nilai keadaan menggunakan teknik heuristik

Finite State
Machine (FSM)

Metodologi Desain
Rangkaian FSM

Implementasi
dengan Flip-flop
Lain

Penyederhanaan
Rangkaian FSM

Permasalahan Pemberian
Nilai Keadaan

Petunjuk Pemberian Nilai
Keadaan

One Hot Encoding

FSM Mealy

Ringkasan

Lisensi

Petunjuk Pemberian Nilai Keadaan

- ▶ Petunjuk ini tidak menjamin solusi yang minimal (paling sederhana)
- ▶ Nilai keadaan-keadaan disebut berdekatan jika mereka hanya berbeda satu variabel keadaan

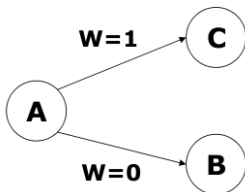
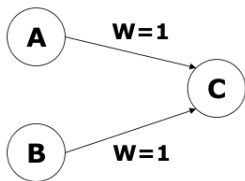
Petunjuk pemberian nilai keadaan:

1. Keadaan-keadaan yang mempunyai `next_state` yang sama untuk suatu masukan **w** **seharusnya** diberikan nilai yang berdekatan
2. Keadaan-keadaan yang merupakan `next_state` dari state yang sama **seharusnya** diberikan nilai yang berdekatan
3. Keadaan-keadaan yang mempunyai keluaran yang sama untuk suatu masukan **seharusnya** diberikan nilai yang berdekatan (grup 1 di keluaran K-map)

Petunjuk Penugasan Keadaan

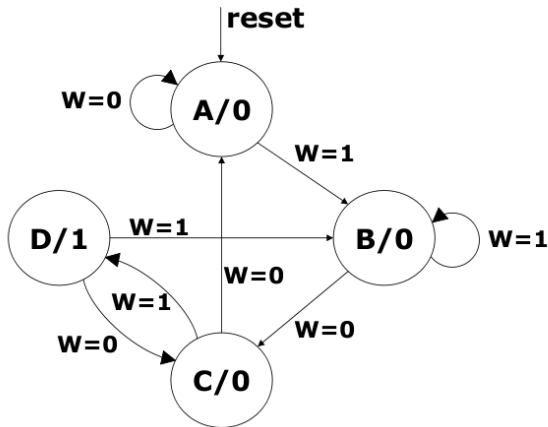
1. Keadaan-keadaan yang mempunyai **next_state** yang sama untuk suatu masukan w **seharusnya** diberikan nilai yang berdekatan

2. Keadaan-keadaan yang merupakan **next_state** dari keadaan yang sama **seharusnya** diberikan nilai yang berdekatan



Contoh Diagram Keadaan Moore

- ▶ Diagram untuk mendeteksi urutan **101**

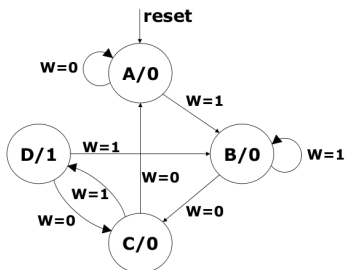


Contoh Pemberian Nilai Keadaan

- ▶ Dua variabel keadaan $y_1 y_0$
- ▶ A=00 (keadaan mulai)

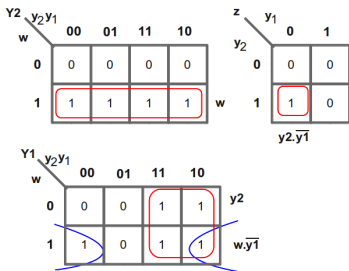
1. {A,D}, {B,D}, {A,C}
2. {A,D}, {B,C}
3. {A,B,C}

$y_1 \backslash y_0$	0	1	
0	A	C	A=00 B=11
1	D	B	C=01 D=10



Tabel Keadaan Bernilai

Present state y_2y_1	Next state Y_2Y_1		Output Z
	$w=0$	$w=1$	
(A) 00	(A) 00	(B) 11	0
(B) 11	(C) 01	(B) 11	0
(C) 01	(A) 00	(D) 10	0
(D) 10	(C) 01	(B) 11	1



- **Tugas:** Gambarkan rangkaian sekuensialnya dan bandingkan dari rangkaian sebelumnya dengan pemberian $A=00$, $B=01$, $C=10$ dan $D=11$

Bahasan

Finite State Machine (FSM)

Metodologi Desain Rangkaian FSM

Kebutuhan Desain

Diagram Keadaan

Tabel dan Variabel Keadaan

Peta Next-state dan Keluaran

Implementasi dan Analisis Rangkaian

Desain Pencacah

Implementasi dengan Flip-flop Lain

Implementasi dengan TFF

Implementasi dengan JKFF

Penyederhanaan Rangkaian FSM

Permasalahan Pemberian Nilai Keadaan

Petunjuk Pemberian Nilai Keadaan

One Hot Encoding

FSM Mealy

Ringkasan

Rangkaian
Sekuensial
Sinkron

@2017, Eko Didik
Widianto (didik@live.undip.ac.id)

Finite State
Machine (FSM)

Metodologi Desain
Rangkaian FSM

Implementasi
dengan Flip-flop
Lain

Penyederhanaan
Rangkaian FSM

Permasalahan Pemberian
Nilai Keadaan

Petunjuk Pemberian Nilai
Keadaan

One Hot Encoding

FSM Mealy

Ringkasan

Lisensi

One Hot Encoding

- ▶ Cara lain untuk memberikan nilai ke keadaan adalah dengan menggunakan variabel keadaan sebanyak jumlah keadaan yang mungkin dalam rangkaian sekuensial tersebut
 - ▶ Tiap keadaan, $(n-1)$ variabel keadaan bernilai 0, sedangkan 1 variabel bernilai 1
 - ▶ Metode ini disebut **one-hot encoding**
 - ▶ Variabel yang bernilai 1 dikatakan 'hot'
 - ▶ Dalam implementasinya, metode ini membutuhkan flip-flop yang lebih banyak, namun akan mempermudah ekspresi keluaran
 - ▶ Ekspresi keluaran yang lebih sederhana akan membuat rangkaian lebih cepat, karena delay propagasi dari keluaran flip-flop ke keluaran berkurang

Desain One Hot Encoding

- ▶ **Tugas:** Desain rangkaian deteksi urutan 101 menggunakan pemberian nilai secara *one-hot encoding*
- ▶ **Solusi.** rangkaian mempunyai 4 keadaan, yaitu A, B, C dan D. Pemberian nilai keadaan secara enkoding *one-hot* dilakukan dengan menyatakan keadaan dalam 4 variabel, yaitu $y_4y_3y_2y_1$, sehingga nilai $A = 0001$, $B = 0010$, $C = 0100$ dan $D = 1000$.

State	Valuasi	Present state $y_4y_3y_2y_1$	Next state $Y_4 Y_3 Y_2 Y_1$		Output Z
			$w=0$	$w=1$	
A	0001	(A) 0001	(A) 0001	(B) 0010	0
B	0010	(B) 0010	(C) 0100	(B) 0010	0
C	0100	(C) 0100	(A) 0001	(D) 1000	0
D	1000	(D) 1000	(C) 0100	(B) 0010	1

- ▶ Fungsi keluaran z lebih sederhana, namun memerlukan 4 buah DFF

FSM Mealy

Rangkaian
Sekuensial
Sinkron

@2017,Eko Didik
Widianto (di-
dik@live.undip.ac.id)

Finite State
Machine (FSM)

Metodologi Desain
Rangkaian FSM

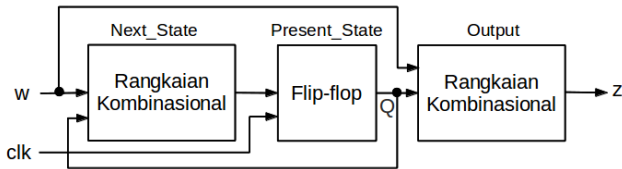
Implementasi
dengan Flip-flop
Lain

Penyederhanaan
Rangkaian FSM

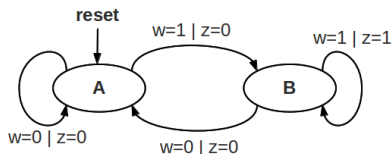
FSM Mealy

Ringkasan

Lisensi



Rangkaian Detektor 1 → 1



► Tabel Keadaan Bernilai

State	Valuasi
A	0
B	1

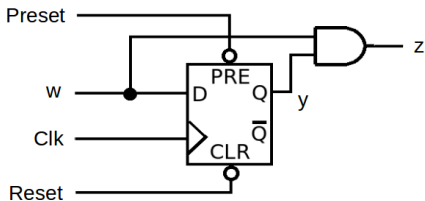
Present state y	Next state, Y		Output, z	
	w=0	w=1	w=0	w=1
0	0	1	0	0
1	0	1	0	1

► Persamaan:

$$Y = w$$

$$z = y \cdot w$$

Rangkaian Detektor



- ▶ Rangkaian hanya membutuhkan 1 DFF dan 1 AND-2 sehingga lebih sederhana daripada rangkaian Moore

- ▶ Yang telah kita pelajari hari ini:
 - ▶ FSM Mesin Moore dan Mealy
 - ▶ Kedua mesin mempunyai rangkaian *next_state*, rangkaian *present_state* (keadaan saat ini) dan rangkaian *output* (keluaran)
 - ▶ Desain rangkaian sekuensial sinkron menggunakan DFF, TFF dan JKFF
 - ▶ Tabel keadaan, tabel keadaan bernilai, tabel transisi dan tabel eksitasi
 - ▶ Desain rangkaian sekuensial lebih sederhana
 - ▶ Desain mesin Mealy dengan DFF

Lisensi

Creative Common Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)

- ▶ Anda bebas:
 - ▶ untuk **Membagikan** — untuk menyalin, mendistribusikan, dan menyebarkan karya, dan
 - ▶ untuk **Remix** — untuk mengadaptasikan karya
- ▶ Di bawah persyaratan berikut:
 - ▶ **Atribusi** — Anda harus memberikan atribusi karya sesuai dengan cara-cara yang diminta oleh pembuat karya tersebut atau pihak yang mengeluarkan lisensi. Atribusi yang dimaksud adalah mencantumkan alamat URL di bawah sebagai sumber.
 - ▶ **Pembagian Serupa** — Jika Anda mengubah, menambah, atau membuat karya lain menggunakan karya ini, Anda hanya boleh menyebarkan karya tersebut hanya dengan lisensi yang sama, serupa, atau kompatibel.
- ▶ Lihat: **Creative Commons Attribution-ShareAlike 3.0 Unported License**

- ▶ Alamat URL: <http://didik.blog.undip.ac.id/buku/sistem-digital/>