

PENGANTAR SISTEM DIGITAL



Analysis of Clocked Sequential Circuits

Dosen Pengajar:

Mitra Pramita S.pd M.pd

Disusun Oleh :

Ahmad Adieb Ibrahim	(1710131210001)
Dimas Aditya P Widya	(1710131310009)
Maidani	(1710131210013)
Muhammad Ridhoni	(1710131310029)
Muhammad Yogi Setiawan	(1610131110005)
Merlina Eris Windia	(1710131120005)
Putri Yani	(1710131220023)

**PROGRAM STUDI PENDIDIKAN ILMU KOMPUTER
FAKULTAS KEGURUAN DAN ILMU PENDIDIKAN
UNIVERSITAS LAMBUNG MANGKURAT**

A. ANALYSIS OF CLOCKED SEQUENTIAL CIRCUITS

Analisis mengilustrasikan apa yang akan dilakukan rangkaian tertentu dalam kondisi operasi tertentu. Urutan perilaku urutan jam ditentukan oleh input, output, dan keadaan flipflop. Output dan status selanjutnya adalah input dan status fungsi saat ini. Analisis sekuensial berurutan terdiri dari mendapatkan tabel atau diagram untuk urutan waktu input, output, dan status internal. Itu juga mungkin untuk menulis ekspresi Boolean yang menggambarkan perilaku rangkaian sekuensial. Ekspresi ini harus mencakup urutan waktu yang diperlukan, baik secara langsung maupun tidak langsung.

Diagram logika dikenali sebagai rangkaian sekuensial *clocked* jika termasuk flip-flop dengan input jam. Flip-flop dapat berupa tipe apa saja, dan diagram logika mungkin atau mungkin tidak termasuk gerbang kombinasional. Pada bagian ini, kami memperkenalkan representasi aljabar untuk menentukan kondisi keadaan-berikutnya dalam hal keadaan saat ini dan input. Tabel keadaan dan diagram keadaan kemudian disajikan untuk menggambarkan perilaku rangkaian sekuensial. Representasi aljabar lain diperkenalkan untuk menentukan diagram logika rangkaian sekuensial. Contoh digunakan untuk menggambarkan berbagai prosedur.

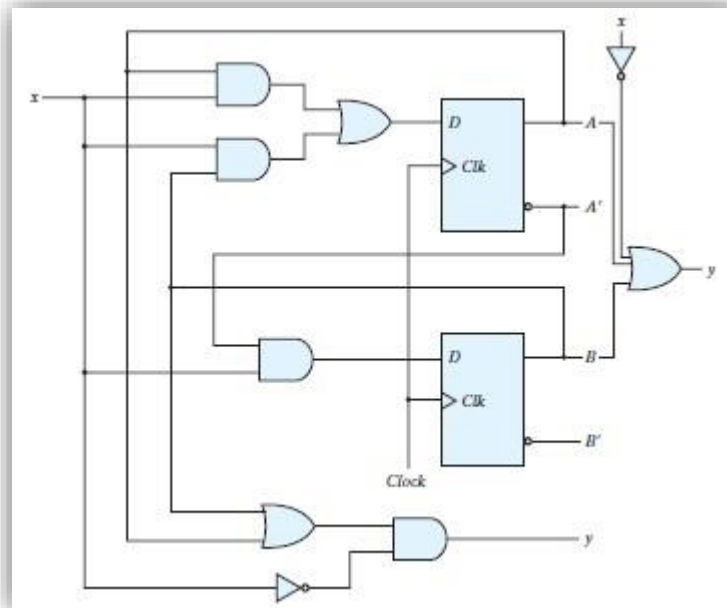
Rangkaian logika *sequential* adalah rangkaian logika yang keadaan outputnya dipengaruhi oleh input dan kondisi rangkaian saat itu. Bentuk dasar dari sekuensial adalah flipflop. Beberapa jenis flip-flop yaitu : flip-flop R-S, flip-flop D, flip-flop J-K, dan flip-flop T.

Unit logika sekuensial atau sering disebut sebagai mesin keadaan berhingga (*finite state machine*, FSM), keluarannya bergantung pada masukan dan keluaran sebelumnya. FSM dibedakan dengan CLU karena selain menghasilkan keluaran juga menghasilkan keadaan (*state*). Hal ini penting untuk implementasi rangkaian memori dan juga unit kendali pada komputer.

1. State Equations

Behavioral rangkaian sekuensial *clock* dapat digambarkan secara aljabar dengan menggunakan persamaan status. Persamaan keadaan (juga disebut persamaan transisi)

menetapkan keadaan selanjutnya sebagai fungsi dari keadaan dan input saat ini. Perhatikan rangkaian sekuensial yang ditunjukkan pada Gambar 1.1. Kami kemudian akan menunjukkan bahwa ia bertindak sebagai 0-detektor dengan menegaskan outputnya kapan



Gambar 1.1

Example of *sequential circuit*

Kami kemudian akan menunjukkan bahwa ia bertindak sebagai 0-detektor dengan menegaskan outputnya ketika sebuah 0 terdeteksi dalam aliran 1s. Ini terdiri dari dua flip flop dan B, sebuah input x dan output y. Karena input D dari flip-flop menentukan nilai status berikutnya (yaitu, state yang dicapai setelah transisi *clock*), adalah mungkin untuk menulis satu set persamaan status untuk rangkaian:

$$A(t+1) = A(t)x(t) + B(t)x(t)$$

$$B(t+1) = A'(t)x(t)$$

Persamaan keadaan adalah ekspresi aljabar yang menentukan kondisi untuk transisi keadaan flip-flop. Sisi kiri persamaan, dengan $(t+1)$, menunjukkan keadaan berikutnya dari flip-flop satu *clock* kemudian. Sisi kanan persamaan adalah ekspresi Boolean yang menentukan keadaan saat ini dan kondisi input yang membuat status berikutnya sama dengan 1. Karena semua variabel dalam ekspresi Boolean adalah fungsi dari keadaan

sekarang, kita dapat menghilangkan penunjukan (t) setelah setiap variabel untuk kenyamanan dan dapat mengekspresikan persamaan status dalam bentuk yang lebih ringkas

$$\mathbf{A(t + 1) = Ax + Bx}$$

$$\mathbf{B(t + 1) = A'x}$$

Ekspresi Boolean untuk persamaan dapat diturunkan langsung dari gerbang yang membentuk bagian sirkuit kombinasional dari rangkaian sekuensial, karena nilai-nilai D dari sirkuit kombinasional menentukan keadaan selanjutnya.

Demikian pula, nilai output saat ini dapat dinyatakan secara aljabar sebagai :

$$\mathbf{y(t) = [A(t) + B(t)]x'(t)}$$

y menghapus simbol (t) untuk keadaan sekarang, kita memperoleh persamaan keluaran Boolean:

$$\mathbf{y = (A + B)x'}$$

2. State Table

Urutan waktu input, output, dan keadaan flip-flop dapat dicacah dalam tabel keadaan (kadang disebut tabel transisi). Tabel ini terdiri dari empat bagian yang diberi label keadaan sekarang, masukan, status berikutnya, dan output. Bagian sekarang-keadaan menunjukkan keadaan flip flop A dan B pada waktu tertentu t. Bagian input memberikan nilai x untuk setiap kemungkinan keadaan sekarang. Bagian berikutnya menunjukkan keadaan flip-flop satu siklus nanti, pada waktu t + 1. Bagian output memberikan nilai y pada waktu t untuk setiap kondisi sekarang dan kondisi input.

Derivasi dari tabel keadaan membutuhkan daftar semua kombinasi biner yang mungkin dari keadaan dan input saat ini. Dalam hal ini, kami memiliki delapan kombinasi biner dari 000 hingga 111. Nilai-nilai selanjutnya-state kemudian ditentukan dari diagram logika atau dari persamaan, Keadaan berikutnya flip-flop A harus memenuhi persamaan keadaan

$$A(t + 1) = Ax + Bx$$

Present State		Input	Next State		Output
A	B		A	B	
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0

Gambar 1.2

State Table for the Circuit

Berikutnya-state bagian tabel di bawah kolom yang memiliki tiga 1 di mana keadaan sekarang dari A dan input x keduanya sama dengan 1 atau keadaan sekarang dari B dan input x keduanya sama dengan 1. Demikian pula, keadaan selanjutnya dari flip-flop B berasal dari persamaan keadaan

$$B(t + 1) = A'x$$

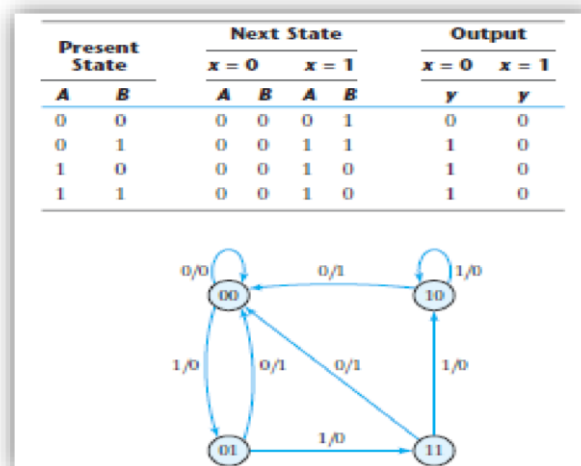
dan sama dengan 1 ketika keadaan sekarang dari A adalah 0 dan input x sama dengan 1. Kolom output diturunkan dari persamaan output

Tabel keadaan *sequential circuits* dengan flip-flop tipe D diperoleh dengan prosedur yang sama yang diuraikan dalam contoh sebelumnya. Secara umum, rangkaian sekuensial dengan m flip-flops dan n input membutuhkan $2m + n$ baris di tabel keadaan. Bilangan biner dari 0 hingga $2m + n - 1$ tercantum di bawah kolom *present-state* dan input. Bagian berikutnya memiliki kolom m, satu untuk setiap flip-flop. Nilai-nilai biner untuk keadaan berikutnya berasal langsung dari persamaan state. Bagian output memiliki banyak kolom karena ada variabel output. Nilai binernya berasal dari rangkaian atau dari fungsi Boolean dengan cara yang sama seperti pada tabel kebenaran.

Kadang-kadang untuk menyatakan tabel keadaan dalam bentuk yang sedikit berbeda hanya memiliki tiga bagian: keadaan sekarang, keadaan berikutnya, dan output. Kondisi input yang disebutkan di bawah bagian bagian dan output berikutnya. Untuk setiap keadaan, ada dua kemungkinan pada status dan output berikutnya, tergantung pada nilai input. Satu formulir mungkin lebih baik dari yang lain, tergantung pada aplikasinya.

3. State Diagram

Informasi yang tersedia di tabel keadaan dapat direpresentasikan secara grafis dalam bentuk diagram keadaan. Dalam jenis diagram ini, keadaan diwakili oleh lingkaran, dan transisi (*clock-triggered*) antar keadaan ditunjukkan oleh garis yang diarahkan menghubungkan lingkaran.



Gambar.3

Second Form of the State Table

Angka biner di dalam setiap lingkaran mengidentifikasi keadaan flip-flop. Garis yang diarahkan diberi label dengan dua bilangan biner yang dipisahkan oleh garis miring. Nilai input selama keadaan sekarang diberi label pertama, dan nomor setelah garis miring memberikan output selama keadaan sekarang dengan input yang diberikan. (Penting untuk diingat bahwa nilai bit yang terdaftar untuk output sepanjang garis yang diarahkan terjadi selama keadaan ini dan dengan input yang ditunjukkan, dan tidak ada hubungannya dengan transisi ke keadaan selanjutnya.) Sebagai contoh, garis yang diarahkan dari menyatakan 00 hingga 01 diberi label 1/0, yang berarti bahwa ketika *sequential circuit* berada dalam keadaan sekarang 00 dan inputnya adalah 1, outputnya adalah 0. Setelah siklus *clock* berikutnya, sirkuit pergi ke keadaan berikutnya, 01. Jika input berubah menjadi 0, maka output menjadi 1, tetapi jika input tetap pada 1, output tetap di 0. Informasi ini diperoleh dari diagram keadaan di sepanjang dua garis

yang diarahkan berasal dari lingkaran dengan keadaan 01. Garis diarahkan menghubungkan lingkaran dengan dirinya sendiri menunjukkan bahwa tidak ada perubahan keadaan terjadi.

Langkah-langkah yang disajikan dalam contoh ini dirangkum di bawah ini:

Sirkuit diagram S Persamaan - Tabel keadaan S Diagram keadaan

Urutan langkahnya dimulai dengan representasi struktural dari rangkaian dan berlanjut ke representasi abstrak dari tindakan. Model HDL dapat berupa deskripsi tingkat gerbang atau dalam bentuk deskripsi behavioral. Penting untuk dicatat bahwa pendekatan tingkat-gerbang mensyaratkan bahwa perancang memahami cara memilih dan menghubungkan gerbang dan flip flop untuk membentuk sirkuit yang memiliki tindakan tertentu. Pemahaman itu datang dengan pengalaman. Di sisi lain, pendekatan berdasarkan pemodelan keadaan tidak mengharuskan perancang untuk mengetahui cara menciptakan skema — perancang hanya perlu mengetahui cara mendeskripsikan behavioral menggunakan konstruksi HDL, karena rangkaian diproduksi secara otomatis oleh suatu sintesis. alat. Oleh karena itu, seseorang tidak harus mengumpulkan pengalaman bertahun-tahun untuk menjadi perancang sirkuit digital yang produktif; juga tidak ada yang harus mendapatkan latar belakang yang luas dalam teknik elektro.

Tidak ada perbedaan antara tabel keadaan dan diagram keadaan, kecuali dalam cara representasi. Tabel keadaan lebih mudah diperoleh dari diagram logika yang diberikan dan persamaan status. Diagram keadaan mengikuti langsung dari tabel keadaan. Diagram keadaan memberikan pandangan bergambar dari transisi dan merupakan bentuk yang lebih cocok untuk interpretasi manusia dari operasi sirkuit. Masukan 0 pertama setelah string 1 memberi output 1 dan mentransfer rangkaian kembali ke keadaan awal, 00. Mesin diwakili oleh diagram keadaan ini bertindak untuk mendeteksi nol dalam aliran bit data.

B. Flip-Flop Input Equations

Diagram logika dari *sequential circuit* terdiri dari flip-flop dan gerbang. Interkoneksi di antara gerbang membentuk sirkuit kombinasional dan dapat ditentukan secara aljabar dengan ekspresi Boolean. Pengetahuan tentang jenis flip-flop dan daftar ekspresi Boolean dari rangkaian kombinasional menyediakan informasi yang diperlukan untuk menggambar diagram

logika dari rangkaian sekuensial. Bagian dari rangkaian kombinasional yang menghasilkan output eksternal dijelaskan secara aljabar dengan satu set fungsi Boolean yang disebut persamaan output. Bagian dari rangkaian yang menghasilkan input ke flip-flop dijelaskan secara aljabar dengan satu set fungsi Boolean yang disebut persamaan input flip-flop (atau, kadang-kadang, persamaan eksitasi). Kami akan mengadopsi konvensi menggunakan simbol masukan flip-flop untuk menunjukkan variabel persamaan input dan subskrip untuk menunjuk nama output flip-flop. Sebagai contoh, persamaan input berikut menetapkan gerbang OR dengan input x dan y terhubung ke input D dari flip-flop yang keluarannya diberi label dengan simbol Q:

$$\mathbf{DQ = x + y}$$

Sequential circuit dari Gambar 1.1 terdiri dari dua D flip-flop A dan B, input x, dan output y. Diagram logika rangkaian dapat dinyatakan secara aljabar dengan dua persamaan input flip-flop dan persamaan output:

$$\mathbf{DA = Ax + Bx}$$

$$\begin{aligned}\mathbf{DB} &= \mathbf{A'x y =} \\ &\mathbf{(A + B)x' }\end{aligned}$$

tiga persamaan menyediakan informasi yang diperlukan untuk menggambar diagram logika dari rangkaian sekuensial. Simbol DA menentukan D-flop berlabel A. DB menentukan D flipflop kedua berlabel B. Ekspresi Boolean yang terkait dengan kedua variabel dan ekspresi untuk output y menentukan bagian sirkuit kombinasional dari *sequential circuit*.

Persamaan masukan flip-flop merupakan bentuk aljabar yang nyaman untuk menentukan diagram logika dari *sequential circuit*. Mereka menyiratkan jenis flip-flop dari simbol huruf, dan mereka sepenuhnya menentukan sirkuit kombinasional yang menggerakkan flip-flop. Perhatikan bahwa ekspresi untuk persamaan input untuk D flip-flop identik dengan ekspresi untuk persamaan status yang sesuai. Ini karena persamaan karakteristik yang menyamakan keadaan berikutnya dengan nilai input D : $\mathbf{Q(t + 1) = DQ}$.

C. Analysis with D Flip-Flops

$$D_A = A \oplus x \oplus y$$

Simbol D_A menyiratkan flip-flop D dengan keluaran A. Variabel x dan y adalah input ke rangkaian. Tidak ada persamaan output yang diberikan, yang menyiratkan bahwa output berasal dari output flip-flop.

Tabel kebenaran memiliki satu kolom untuk keadaan sekarang dari flip-flop A, dua kolom untuk dua input, dan satu kolom untuk status A berikutnya. Nilai-nilai keadaan berikutnya diperoleh dari persamaan keadaan :

$$A(t + 1) = A \oplus x \oplus y$$

Ekspresi menentukan fungsi ganjil dan sama dengan 1 ketika hanya satu variabel adalah 1 atau ketika ketiga variabel adalah 1. Ini ditunjukkan dalam kolom untuk status A berikutnya. Sirkuitnya memiliki satu flip-flop dan dua bagian states. Keadaan sekarang dan output dapat berupa 0 atau 1, seperti yang ditunjukkan pada gambar 5.17. Garis miring pada garis yang diarahkan tidak diperlukan, karena tidak ada output dari sirkuit kombinasional. Kedua input dapat memiliki empat kemungkinan kombinasi untuk setiap bagian state. Dua kombinasi input selama setiap transisi status dipisahkan oleh koma untuk menyederhanakan notasi.

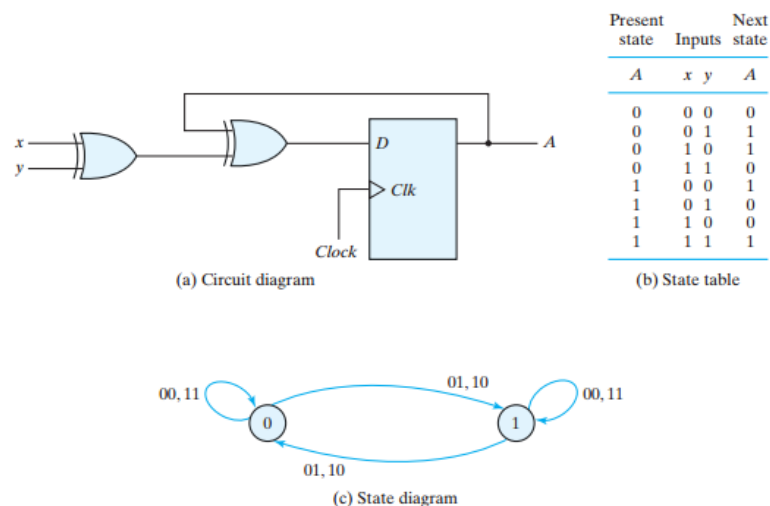


FIGURE 5.17
Sequential circuit with D flip-flop

D. Flip-flop JK dan T

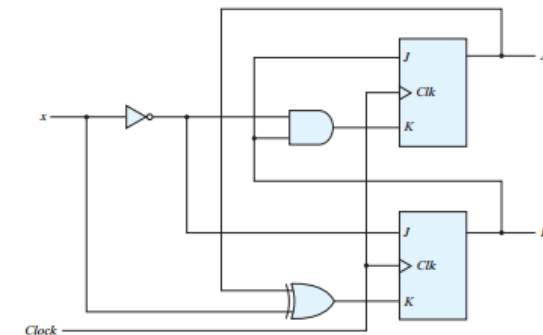


FIGURE 5.18
Sequential circuit with JK flip-flop

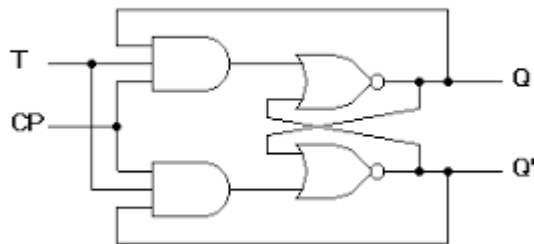
Tabel keadaan terdiri dari empat bagian: keadaan sekarang, input, status berikutnya, dan output. Dua yang pertama diperoleh dengan mencantumkan semua kombinasi biner. Bagian output ditentukan dari persamaan output. Nilai-nilai keadaan berikutnya dievaluasi dari persamaan negara. Untuk D-type flip-flop, persamaan keadaan sama dengan persamaan input. Ketika sebuah flip-flop selain tipe D digunakan, seperti JK atau T, perlu untuk merujuk ke tabel karakteristik yang sesuai atau persamaan karakteristik untuk mendapatkan nilai-nilai status berikutnya. Kami akan menggambarkan prosedur pertama dengan menggunakan tabel karakteristik dan lagi dengan menggunakan persamaan karakteristik. Nilai-nilai selanjutnya dari rangkaian sekuensial yang menggunakan flip-flop tipe JK atau T-type, yaitu sebagai berikut:

1. Tentukan persamaan input flip-flop dalam hal keadaan saat ini dan masukan variabel.
2. Buat daftar nilai-nilai biner dari setiap persamaan input.
3. Gunakan tabel karakteristik flip-flop yang sesuai untuk menentukan keadaan selanjutnya nilai-nilai di tabel state. Sebagai contoh, perhatikan rangkaian sekuensial dengan dua flip-flop JK A dan B dan satu input x, seperti ditunjukkan pada Gambar. 5.18. Sirkuit tidak memiliki output; oleh karena itu, tabel state tidak membutuhkan kolom keluaran. (Output dari flip-flop dapat dianggap sebagai output dalam hal ini.) Rangkaian dapat ditentukan oleh persamaan input flip-flop.

$$J_A = B \quad K_A = Bx'$$

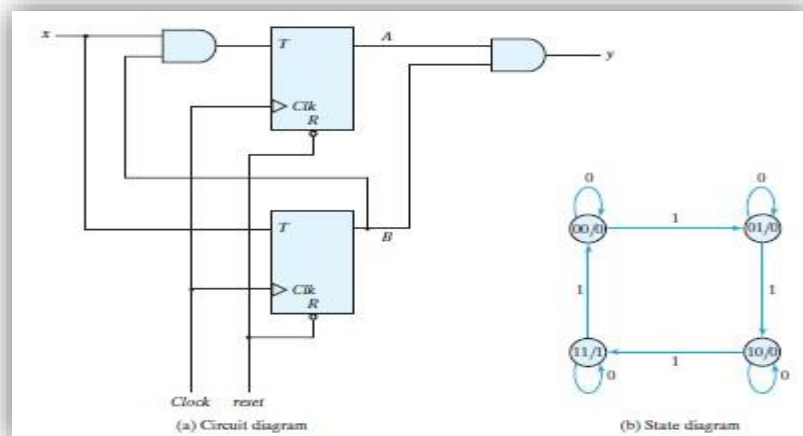
$$J_B = x' \quad K_B = A'x + Ax' = A \oplus x$$

T Flip-flop merupakan rangkaian flip-flop yang telah di buat dengan menggunakan flip-flop J-K yang kedua inputnya dihubungkan menjadi satu maka akan diperoleh flip-flop yang memiliki watak membalik output sebelumnya jika inputannya tinggi dan outputnya akan tetap jika inputnya rendah. Berikut adalah gambar tabel kebenaran gerbang logika dan symbol dari T Flip - flop.



E. Mealy and Moore Models of Finite State Machines

Model sirkuit sekuensial yang paling umum memiliki input, output, dan status internal. Ini adalah kebiasaan untuk membedakan antara dua model rangkaian sekuensial : model *Mealy* dan model *Moore*. Keduanya ditunjukkan pada gambar, dan mereka hanya berbeda dalam hal output dihasilkan.



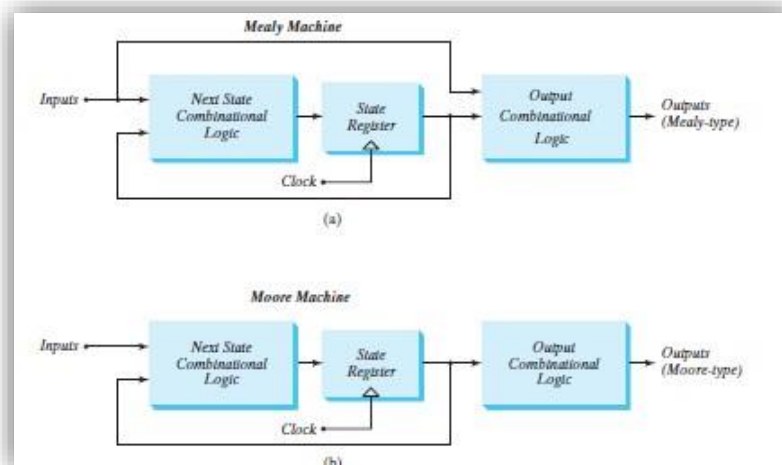
Gambar 1.7

Sequential circuit with T flip-flops (Binary-Counter)

Present State		Input	Next State		Output
A	B		A	B	
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	1	0	0
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	1

Gambar 1.8

Table for Sequential circuit with T Flip-Flops



Gambar 1.9

Block diagrams of Mealy and Moore state machines

Dalam model *Mealy*, output adalah fungsi dari kedua keadaan saat ini dan masukan. Dalam model *Moore*, output adalah fungsi dari hanya keadaan sekarang. Sebuah sirkuit mungkin memiliki kedua jenis output. Dua model rangkaian sekuensial umumnya disebut

sebagai state machine yang terbatas, disingkat FSM. Model *Mealy* yang berurutan Rangkaian ini disebut sebagai mesin *Mealy* FSM atau *Mealy*. Model *Moore* disebut sebagai *Moore* FSM atau mesin *Moore*. Keluaran y adalah fungsi dari kedua input x dan keadaan A dan B saat ini. Outputnya adalah keadaan flip-flop yang ditandai di dalam lingkaran. Hasil hanya bergantung pada nilai-nilai flip-flop, dan itu membuatnya menjadi fungsi dari keadaan sekarang. Nilai input dalam diagram states diberi label sepanjang garis yang diarahkan, tetapi hasilnya nilai diindikasikan di dalam lingkaran bersama dengan keadaan sekarang. Dalam model *Moore*, output dari rangkaian sekuensial disinkronkan dengan *clock*, karena mereka hanya bergantung pada output flip-flop yang disinkronkan dengan *clock*. Dalam model *Mealy*, output dapat berubah jika input berubah selama siklus *clock*. Selain itu, output mungkin memiliki nilai palsu sesaat karena penundaan ditemui dari waktu ketika input berubah dan waktu ketika output flip-flop berubah. Untuk menyinkronkan sirkuit tipe *Mealy*, input dari rangkaian sekuensial harus disinkronkan dengan *clock* dan hasilnya harus diambil sampelnya segera sebelum tepi *clock*. Input diubah pada tepi tidak aktif dari *clock* untuk memastikan bahwa input ke flip-flop stabil sebelum tepi aktif *clock* terjadi. Demikian, output mesin *Mealy* adalah nilai yang hadir segera sebelum tepi *clock* aktif.

DAFTAR PUSTAKA

Mano, M. M., & Ciletti, M. D. (2013). Digital Design With an introduction to the Verilog HDL.

Suyanto, Yohanes. (2009). Perancangan Sistem Digital.