

FA dengan Output

- ▶ FSA : accepter, dapat menerima atau tidak.
- ▶ FSA dengan output : transducer
 1. Mesin Moore :output berasosiasi dengan state
 2. Mesin Mealy :output berasosiasi dengan transisi

DEFINISI dan IMPLEMENTASI MESIN MOORE

- ▶ Mesin Moore didefinisikan dalam 6(enam) Tupple, $M_0 = (Q, \Sigma, \delta, S, \Delta, \lambda)$, dimana :
 - Q = Himpunan state terbatas / letter
 - Σ = Himpunan symbol input
 - δ = Fungsi Transisi
 - S = State awal, $S \in Q$
 - Δ = Himpunan symbol output / karakter
 - λ = Fungsi output untuk setiap state



- ▶ Jikalau input string misalnya n abjad, maka output string akan terdapat n+1 karakter karena terdapat n+1 state yang dilalui dalam prosesnya / path.




Representasi Graf


- ▶ Misal kita ingin memperoleh sisa pembagian (modulus) suatu bilangan dengan 3.
- ▶ Dengan input yang dinyatakan dalam biner.
Maka :
 - $Q = (q_0, q_1, q_2)$, $\Sigma = \{0, 1\}$, $\Delta = \{0, 1, 2\}$, $S = q_0$,
 $\lambda(q_0) = 0$, $\lambda(q_1) = 1$, $\lambda(q_2) = 2$



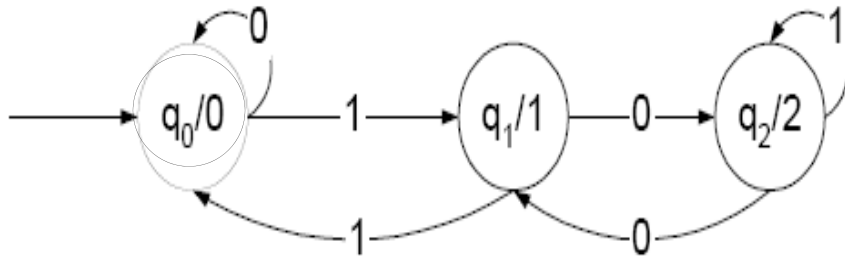
- ▶ Prinsip:
 - jika i diikuti dengan 0, maka hasilnya $2i$
 $101_2 = 5$
 $1010_2 = 2 * 5 = 10$
 - jika i diikuti dengan 1, maka hasilnya $2i + 1$
 $101_2 = 5$
 $1011_2 = 2 * 5 + 1 = 11$



- ▶ jika $i/3$ mempunyai sisa p , dan untuk input berikutnya bernilai 0 maka
 - $2i/3$ mempunyai sisa $2p \bmod 3$
 - untuk $p=0$ maka $2p \bmod 3 = 0$
 - untuk $p=1$ maka $2p \bmod 3 = 2$
 - untuk $p=2$ maka $2p \bmod 3 = 1$
- 

- ▶ jika $i/3$ mempunyai sisa p , dan untuk input berikutnya bernilai 1 maka
 - $(2i+1)/3$ mempunyai sisa $(2p+1) \bmod 3$
 - untuk $p=0$ maka $(2p+1) \bmod 3 = 1$
 - untuk $p=1$ maka $(2p+1) \bmod 3 = 0$
 - untuk $p=2$ maka $(2p+1) \bmod 3 = 2$
- 

- ▶ Sehingga didapat mesin FSA sbb :



Contoh :

input 5 (101_2) , state terakhir $q_2/2$, $5 \bmod 3 = 2$

input 10 (1010_2) , state terakhir $q_1/1$, $10 \bmod 3 = 1$

Mesin Mealy

$M = \{Q, \Sigma, \delta, S, \Delta, \lambda\}$

Q : himpunan state

Σ : himpunan simbol input

δ : fungsi transisi

S : state awal $S \in Q$

Δ : himpunan output

λ : fungsi output untuk setiap transisi

Definisi Mesin MEALY

Mesin Mealy didefinisikan 6(enam) tuple, M_e
 $= \{Q, \Sigma, \delta, S, \Delta, \lambda\}$, dimana :

Q = Himpunan state terbatas / letter



Σ = Himpunan symbol input

δ = Fungsi Transisi

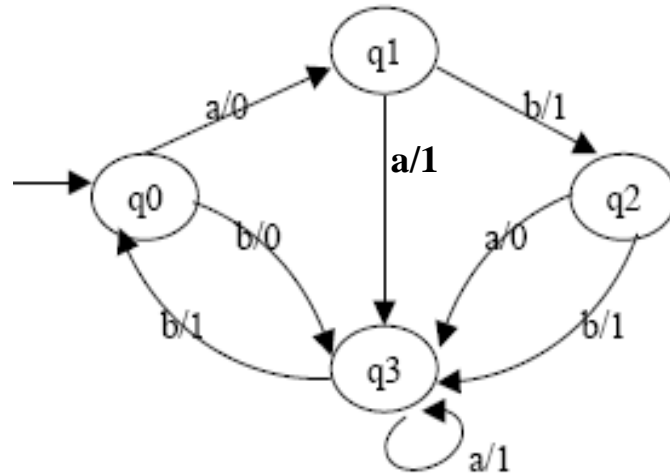
S = State awal, $S \in Q$

Δ = Himpunan symbol output / karakter

λ = Fungsi output untuk setiap transisi

- 
- ▶ Setiap Edge yang terbentuk diberi label komposisi symbol i/o (input/output).
 - ▶ Setiap state harus mempunyai satu buah edge yang mengarah ke state berikutnya untuk setiap kemungkinan input letter.
 - ▶ Edge yang dilalui tergantung kepada input letter i , ketika menjelajahi edge kita harus mencetak output karakter o .
 - ▶ Mesin Mealy tidak mendefinisikan suatu bahasa dalam menerima untai-untai, karena setiap input string menghasilkan output string dan tidak terdapat final state.
 - ▶ Jikalau input string misalnya n abjad, maka output string akan terdapat n karakter.
- 

Misal



input	: aaabb
State yang ditempuh	: q1-q3-q3-q0-q3
Output	: 01110

Contoh mesin Mealy untuk mendeteksi ekspresi reguler $(0+1)^*(00+11)$

Jawab

$M = \{Q, \Sigma, \delta, S, \Delta, \lambda\}$

$Q : \{q0, q1, q2\}$

$\Sigma : \{0, 1\}$

$S : q0$

$\Delta : \{Y, T\}$

$\lambda(q0, 0) = T$

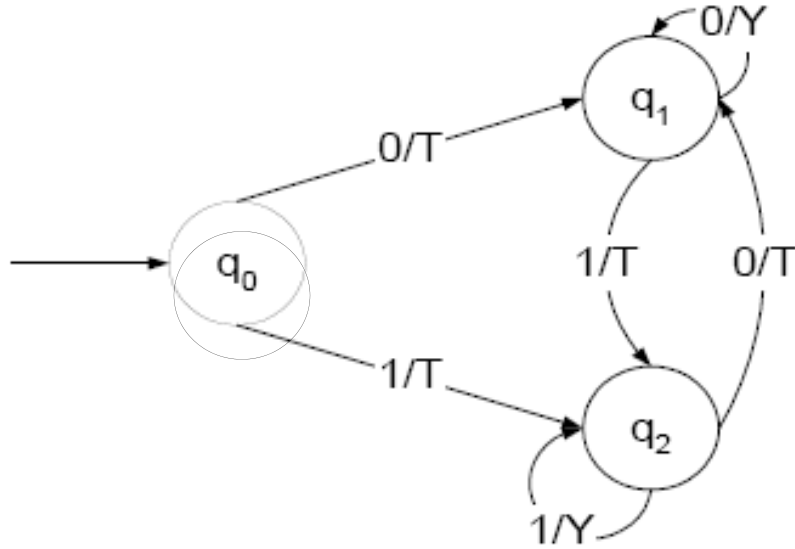
$\lambda(q0, 1) = T$

$\lambda(q1, 0) = Y$

$\lambda(q1, 1) = T$

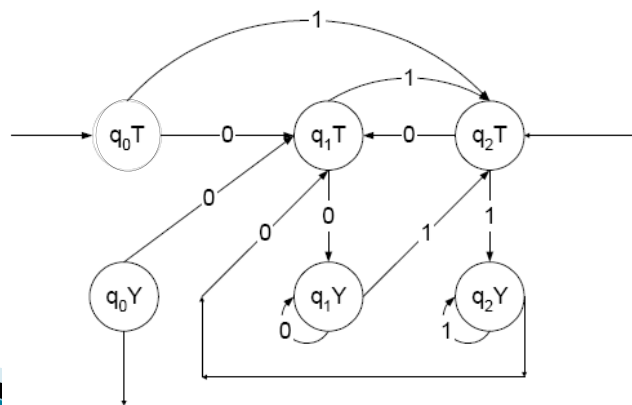
$\lambda(q2, 0) = T$

$\lambda(q2, 1) = Y$



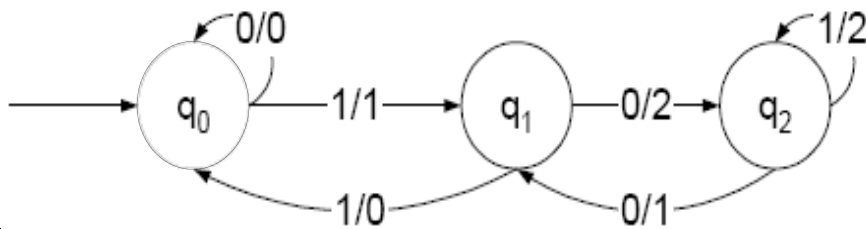
Ekuivalensi mesin Moore dengan mesin Mealy

- ▶ Mesin Moore ke mesin Mealy
Jml state = jml state sebelum * jml output



- ▶ Mesin Mealy ke mesin Moore

Menambah label output pada transisi
Menghapus label output pada state

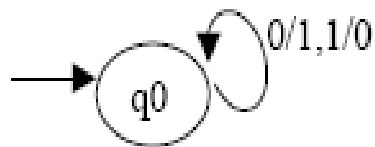


Implementasi Pada Sirkuit Sekuensial

- ▶ Aplikasi Finite Automata dengan output pada sekuensial sirkuit berkaitan erat dengan Arsitektur computer.
- ▶ Penerapan Model Mesin ini umumnya berfungsi sebagai penjelas dari aksi sekuensial sirkuit yang dimodelkan

Contoh:

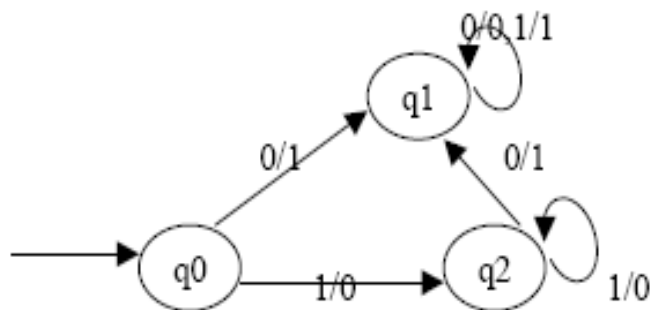
- ▶ Mesin penghasil 1's komplemen yang bisa dievolusikan menjadi mesin pengurangan



input : 001010

output : 110101

- ▶ Incrementer yang biasa kita temukan pada CPU

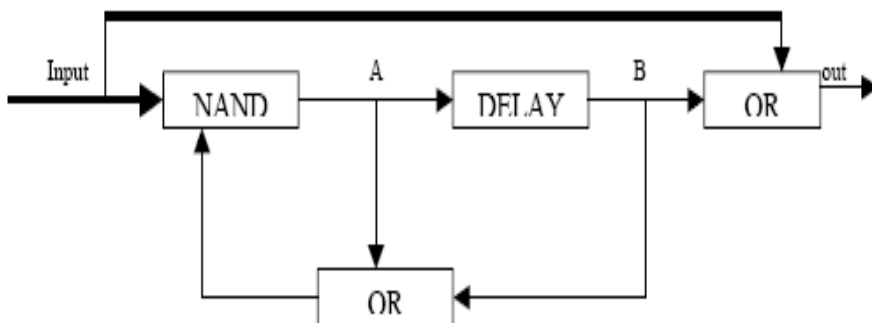


input : 1011 -> dibaca terbalik 1101

output : 0011 -> ditulis terbalik 1100

- ▶ Flip-flop merupakan suatu rangkaian gerbang logika yang mempunyai dua keadaan stabil pada outputnya, dalam hal ini direpresentasikan dengan bit 0,1.
- ▶ Terdapat beberapa jenis Flip-flop yaitu : SR Flip-Flop, JK Flip-flop, Master-Slave JKFF, dan D Flip-flop.
- ▶ Flip-flop berfungsi menyimpan data

- ▶ D Flip Flop



Diketahui terdapat 4(empat) state berdasarkan ada tidaknya arus pada titik A dan B di sirkuit

Q0 adalah $A=0$ $B=0$

Q1 adalah $A=0$ $B=1$

Q2 adalah $A=1$ $B=0$

Q3 adalah $A=1$ $B=1$



Rule perubahan state setelah input 0 atau 1

New B = Old A

New A = (input) NAND (Old A or Old B)

Output = (input) OR (old B)



Pada pulsa diskrit yang bervariasi pada penerimaan input

Time clock, state berubah dan outputpun dihasilkan.

Misalkan kita berada pada state q_0 dan menerima input 0 :

$$\begin{aligned} \text{New B} &= \text{Old A} = 0 \\ \text{New A} &= (\text{input}) \text{ NAND } (\text{Old A or Old B}) \\ &= (0) \text{ NAND } (0 \text{ OR } 0) \\ &= 0 \text{ NAND } 0 \\ &= 1 \end{aligned}$$

$$\text{Output} = 0 \text{ or } 0 = 0$$

State baru adalah q_2 (karena $\text{New A}=1$, $\text{New B}=0$).



Jika kita berada pada q_0 dan menerima input 1 :

$$\begin{aligned} \text{New B} &= \text{Old A} = 0 \\ \text{New A} &= 1 \text{ NAND } (0 \text{ OR } 0) = 1 \\ \text{Output} &= 1 \text{ OR } 0 = 1 \end{aligned}$$

State baru adalah q_2 (karena $\text{New A}=1$, $\text{New B}=0$)



Jika kita di q1 dan menerima input 0 :

$$\text{New B} = \text{Old A} = 0$$

$$\text{New A} = 0 \text{ NAND } (0 \text{ OR } 1) = 1$$

$$\text{Output} = 0 \text{ OR } 1 = 1$$

State baru adalah q2



Jika kita berada pada q1 dan menerima input 1:

$$\text{New B} = \text{Old A} = 0$$

$$\text{New A} = 1 \text{ NAND } (0 \text{ OR } 0) = 1$$

$$\text{Output} = 1 \text{ OR } 1 = 1$$

State baru adalah q0



Jika kita berada pada q2 dan menerima input 0:

$$\text{New B} = \text{Old A} = 1$$

$$\text{New A} = 0 \text{ NAND } (1 \text{ OR } 0) = 1$$

$$\text{Output} = 0 \text{ OR } 0 = 0$$

State baru adalah q3 (karena New A=1, New B=1)



Jika kita berada pada q2 dan menerima input 1:

$$\text{New B} = \text{Old A} = 1$$

$$\text{New A} = 1 \text{ NAND } (1 \text{ OR } 0) = 0$$

$$\text{Output} = 1 \text{ OR } 0 = 1$$

State baru adalah q1



Jika kita berada pada q3 dan menerima input 0:

$$\text{New B} = \text{Old A} = 0$$

$$\text{New A} = 0 \text{ NAND } (1 \text{ OR } 1) = 1$$

$$\text{Output} = 0 \text{ OR } 1 = 1$$

State baru adalah q3



Jika kita berada pada q3 dan menerima input 1:

$$\text{New B} = \text{Old A} = 1$$

$$\text{New A} = 1 \text{ NAND } (1 \text{ OR } 1) = 0$$

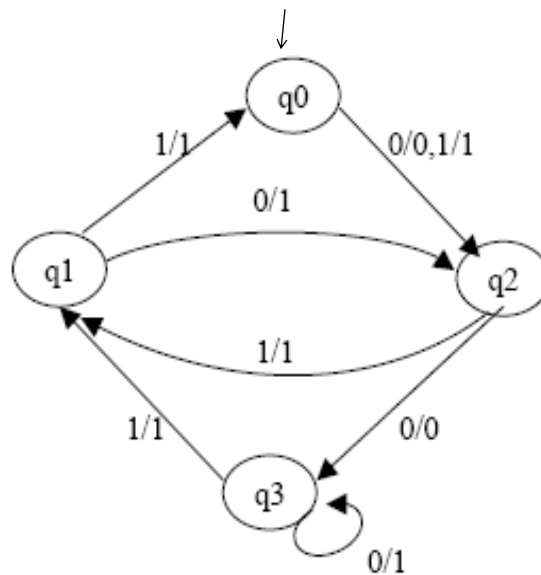
$$\text{Output} = 1 \text{ OR } 1 = 1$$

State baru adalah q1



Old state	Setelah input 0		Setelah input 1	
	New state	Output	New state	Output
q0	q2	0	q2	1
q1	q2	1	q0	1
q2	q3	0	q1	1
q3	q3	1	q1	1

- Representasi aksi dari sirkuit tersebut dalam Mesin Mealy



- ▶ Finite State Automata dengan output merupakan Evolusi dari Finite State Automata konvensional yang mempunyai kemampuan lebih yaitu menghasilkan output.
- ▶ Mesin Mealy dan Mesin Moore merupakan contoh konkret dari FSA dengan output.
- ▶ Tujuan inti dari penemu Mesin ini adalah untuk mendisain suatu model matematika untuk sirkuit sekuensial



- ▶ Output pada mesin Mealy berasosiasi dengan Rule transisi sementara pada Mesin Moore Output berasosiasi dengan state.
- ▶ Pada Mesin Mealy maupun Moore tidak terdapat Final state sehingga tidak bisa mendefinisikan suatu bahasa dari untai-untai masukan

