

# Mata Kuliah

# Dasar Teknik Digital

# TKE 113



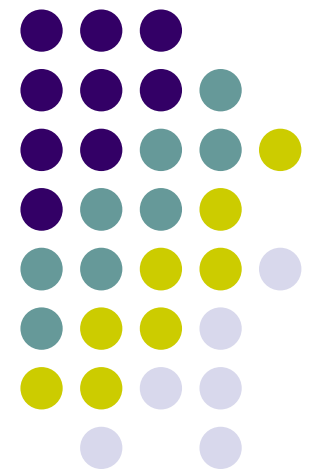
## 8. REGISTER

Ir. Pernantin, M.Sc

Fahmi, S.T, M.Sc

Departemen Teknik Elektro  
Universitas Sumatera Utara USU

2006



# Definisi



- sekumpulan sel biner yang dipakai untuk menyimpan informasi yang disajikan dalam bentuk kode biner
- Dilakukan melalui penyetelan keadaan kumpulan flip-flop dalam register secara serentak sebagai satu kesatuan.

1 Flip flop = 1 bit

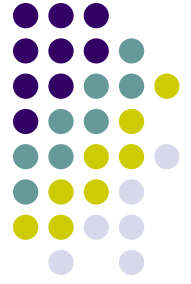
Register 8 bit = data 0 s.d 255 desimal

# Register Pemalang



- Data yang diberikan pada masukan disimpan dan dipalang di dalam register. Setelah pemalangan terjadi, keadaan keluaran register tidak akan berubah walaupun masukannya berubah, berfungsi sebagai penyangga (buffer).
- tdd 2 jenis : *transparan (transparent)* dan *terpicu (triggered)*.
- biasanya dipakai flip-flop D

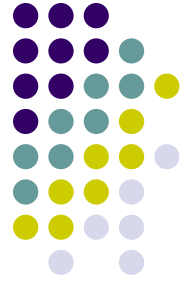
# Register Pemalang [2]



- Transparent
  - perubahan keluaran terjadi pada saat penabuh level high
  - pemalangan terjadi pada saat penabuh pada level low

CP	D	Q <sub>G</sub>
H	H	H
H	L	L
L	x	Q <sub>0</sub>

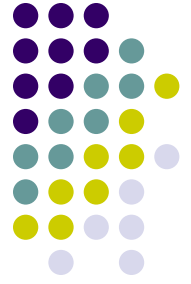
# Register Pemalang [3]



- Triggered
  - perubahan keluaran terjadi pada saat penabuh berubah dari level high ke level low rendah
  - pemalangan terjadi saat penabuh level low

CP	D	Q <sub>T</sub>
↓	H	H
↓	L	L
L	x	Q <sub>0</sub>

# Register Pemalang [4]



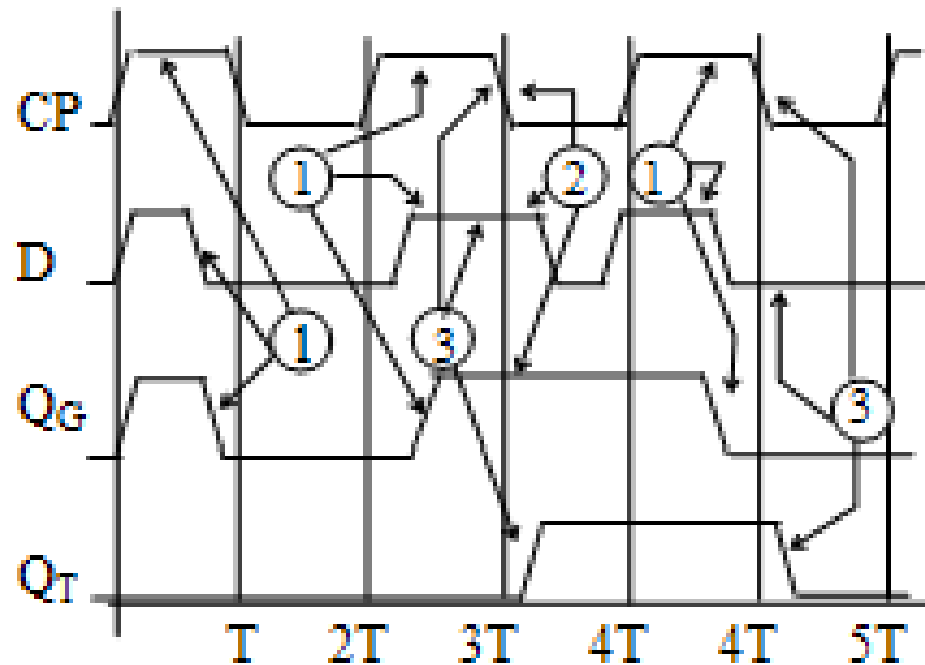
- Contoh time line :

CP = clock pulse,

D = input flip flop D

QG = Keluaran Register pemalang tranparent

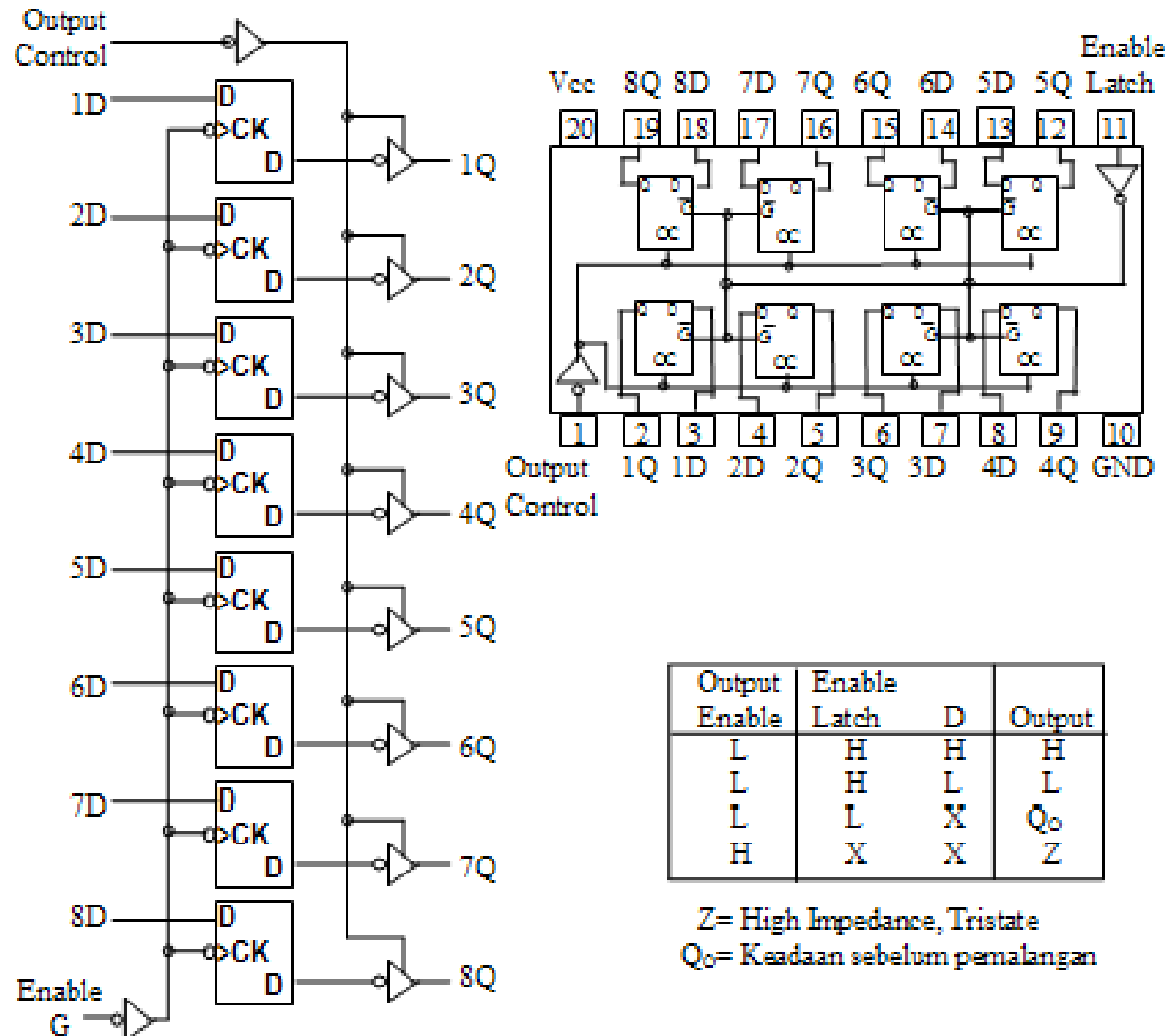
QT = Keluaran Register pemalang triggered



# Register Pemalang [5]



IC 74LS373



# Memori



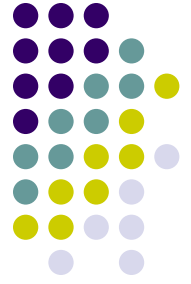
- RAM → sekumpulan flip-flop D
- Masing-masing flip-flop dikenal berdasarkan nomor alamat (address)-nya. →  $A_i$
- Sinyal pemilih :: menentukan sel yang akan ditulis ( $R_i$ ) atau dibaca ( $W_i$ )

Hubungan logika sbb :

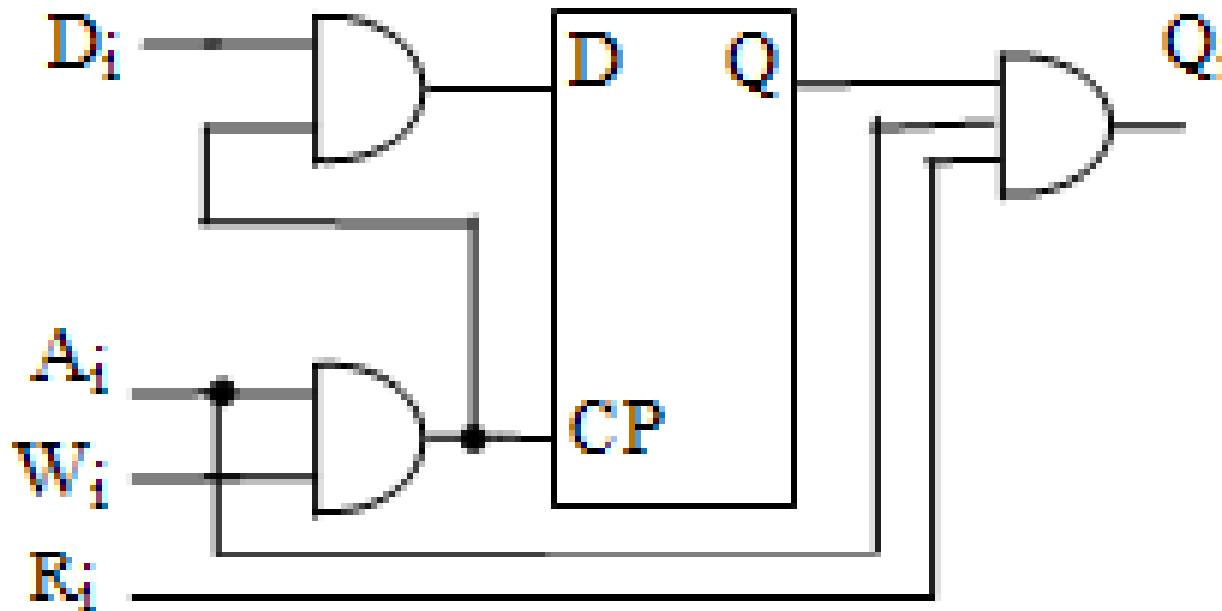
- Penabuh:  $CP = A_i W_i$
- Data masukan:  $D = D_i CP$
- Data Keluaran:  $Q_i = A_i R_i Q$



# Memori [2]



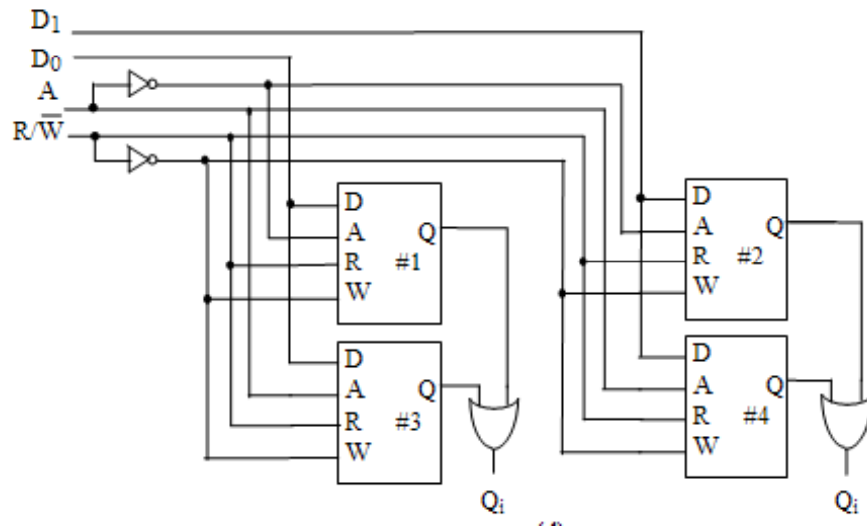
- Rangkaian Logika Memori



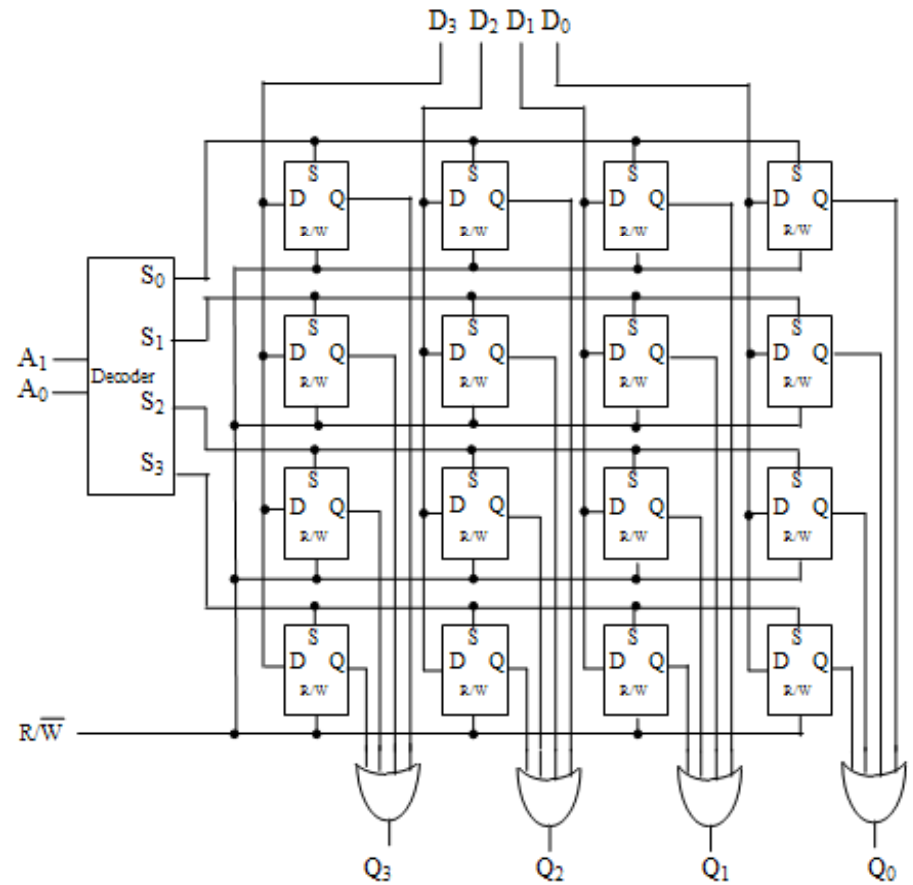
# Memori [3]



- RAM 2x2



## RAM 4x4



# Register Geser Seri



- Kegunaan : komunikasi data serial – paralel, algoritma perkalian biner

- Geser kanan:

Sebelum penggeseran: 1 0 0 1 1 0 1 0

Geser 1 x : 0 1 0 0 1 1 0 1

Geser 2 x : 0 0 1 0 0 1 1 0

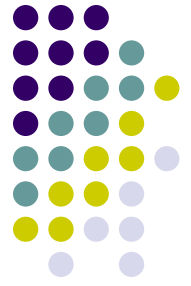
- Geser kiri:

Sebelum penggeseran: 1 0 0 1 1 0 1 0

Geser 1 x : 0 0 1 1 0 1 0 0

Geser 2 x : 0 1 1 0 1 0 0 0

# Register Geser Seri [2]

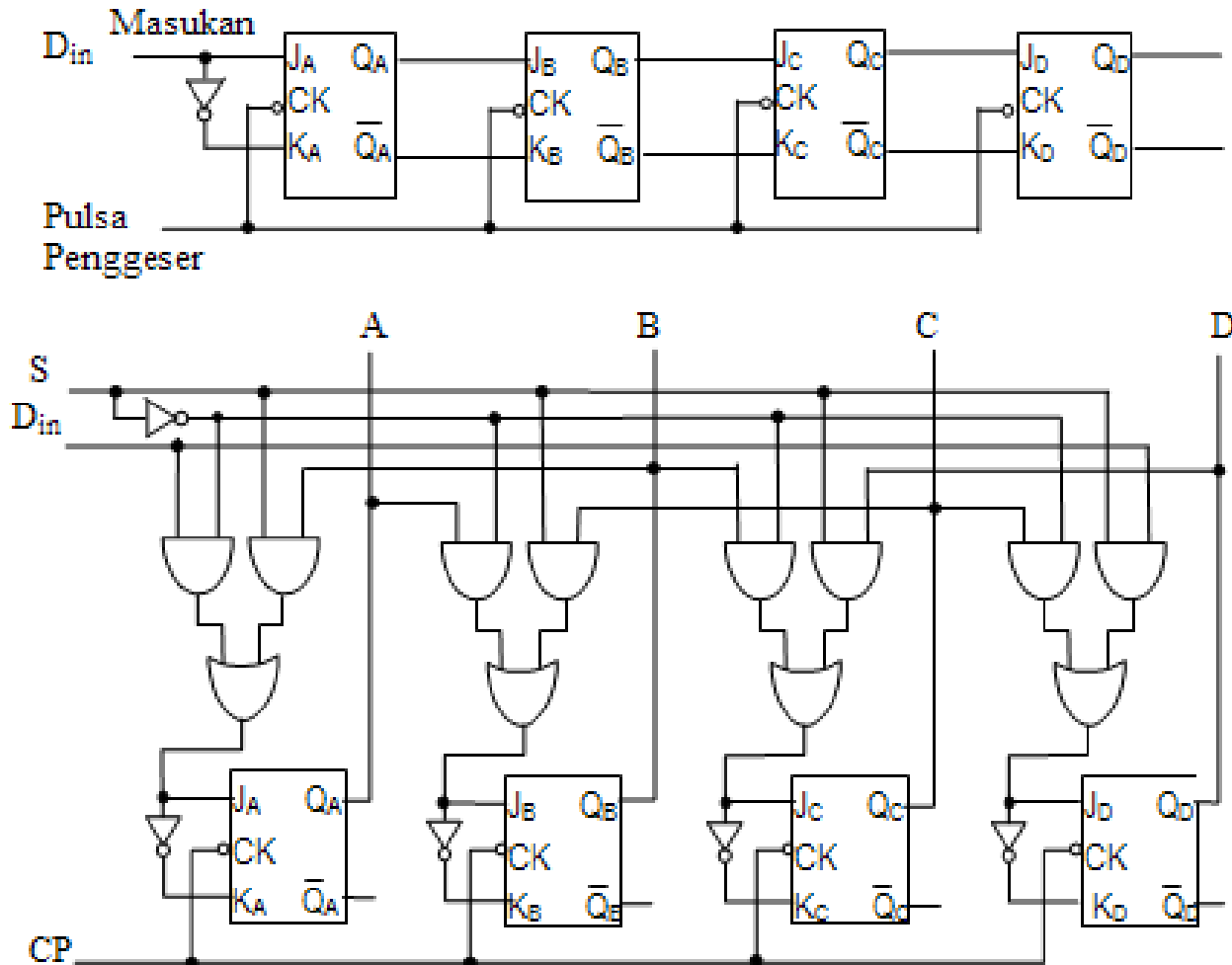


- Realisasi : keluaran satu flip-flop diberikan kepada masukan flip-flop berikutnya dalam urutan penggeseran
- Contoh dengan flip-flop JK 4 bit register geser:  
JA = Din    KA = JA    JC = QB    KC = JC  
JB = QA    KB = JB    JD = QC    KD = JD
- masukan K = J → flip flop D
- Din = masukan luar untuk mengganti bit ujung

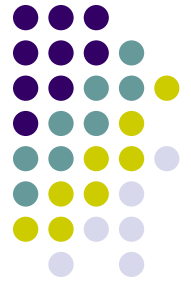
# Register Geser Seri [3]



- Rangkaian Logika (atas), bidirectional (bawah)



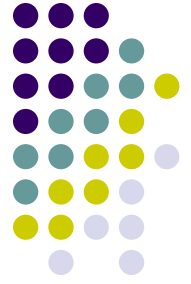
# Register Geser Paralel



- Kegunaan : komunikasi data paralel 1 word = 8 bit
- Modifikasi register geser seri
- masing-masing flip-flop : 3 masukan
  - keluaran flip-flop di kiri (geser kanan)
  - keluaran flip-flop di kanan (geser kiri)
  - masukan paralel dari luar
- Modus Operasi :

Pemilih		Operasi
S1	S0	
0	0	geser kanan
0	1	geser kiri
1	0	muat
1	1	diam

# Register Geser Paralel [2]



- Realisasi dengan Flip Flop RS

$$S_A = \overline{S_1} \overline{S_0} D_{in} + \overline{S_1} S_0 Q_B + S_1 \overline{S_0} A \quad R_A = \overline{S_A}$$

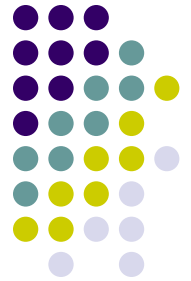
$$S_B = \overline{S_1} \overline{S_0} Q_A + \overline{S_1} S_0 Q_C + S_1 \overline{S_0} B \quad R_B = \overline{S_B}$$

$$S_C = \overline{S_1} \overline{S_0} Q_B + \overline{S_1} S_0 Q_D + S_1 \overline{S_0} C \quad R_C = \overline{S_C}$$

$$S_D = \overline{S_1} \overline{S_0} Q_C + \overline{S_1} S_0 D_{in} + S_1 \overline{S_0} D \quad R_D = \overline{S_D}$$

$$\text{Penabuh} = \overline{S_1 S_0} CP$$

# Register Geser Paralel [3]



- Rangkaian Logika

