# Digital Teknik II / Digital Design II

Dr. D. M. Akbar Hussain
Department of Electronic Systems



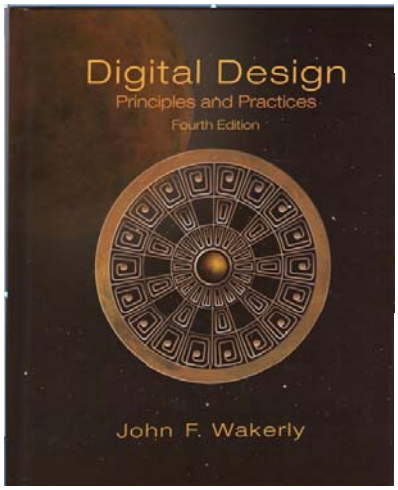
## Digital Teknik / Digital Design

**Underviser:** D. M. Akbar Hussain

**Litteratur:** Digital Design Principles & Practices
4th Edition by John F. Wakerly

Dr. D. M. Akbar Hussain
Department of Electronic Systems

# Course Book






# Course Web Page

http://www.aue.auc.dk/~akbar/2010/Digital-Design-2-2010.html

## Basic Aim (Hovedformålet)

To enable students to apply analysis, synthesis and implementation of basic digital circuits.



## Objective (Mål)

Students can make analysis and should be able to design digital circuits which is a central feature of data or electrical engineering.

## Course Contents (Kursets Indhold)

- Multi-vibrators & Sequential circuits.
- Bi-stable Circuits, Structure of Mono-stable and A-stable circuits.
- Modeling , Analysis and Synthesizing of circuits.
- Mealy and Moore State Machines there analysis, design and realization.
- Counters and Shift Registers.



## Digital Teknik / Digital Design

Modul 1

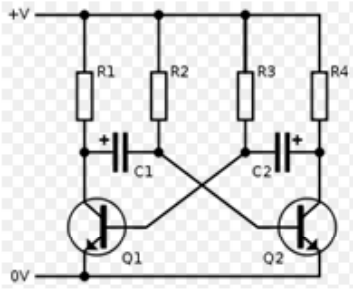
# Multivibrator Circuit

A circuit which is used to build most common two state systems, for example Oscillator, Timers and flip-flops.

- **Astable:** Circuit is not stable in either state and continuously oscillates from one state to the other and it does not require any input, for example clock etc.
- **Monostable:** Obviously as the name says it can have one stable state and one unstable state and it can flip into the unstable state for a determined period, but will eventually return to the stable state. Such a circuit is useful for creating a timing period of fixed duration in response to some external event. This circuit is also known as a one shot.
- **Bistable:** In such a circuit it will remain in one of the two states indefinitely. The circuit can be flipped from one state to the other by an external event or trigger. This type circuit is the fundamental building block of a register or memory device. This circuit is also known as a latch or a flip-flop.



# Multivibrator Circuit



# Sequential Circuit

**Depend upon current input values as well as past sequence applied .**

**A circuit with n binary state variables can have $2^n$ possible states.**

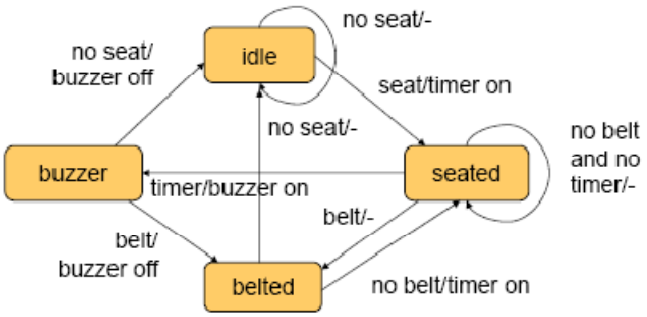
- **Always Finite**
- **Never Infinite**



# Some Basic Stuff

- What is a FSM
- It is a quintuple (Σ, S, A, δ, F), where:
- Σ input alphabet (a finite set of symbols)
- S finite set of states
- A an initial state, an element of S
- δ the state transition function: δ: S x Σ → S
- F set of final states, a subset of S.

# Example



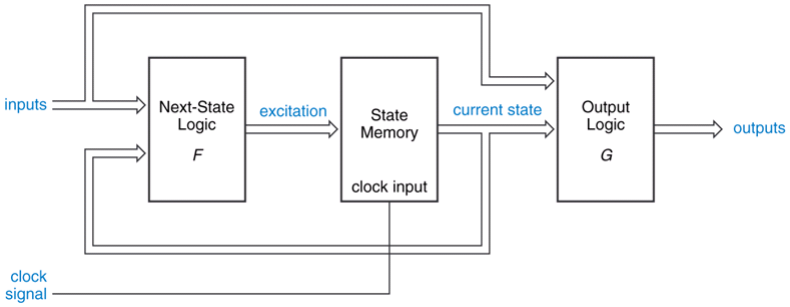



# FSM Models

**Mealy & Moore Models**

**Input Actions (State & Inputs)**
**Entry Actions (State)**

# Structure of a State Machine




Clocked Synchronous State Machine

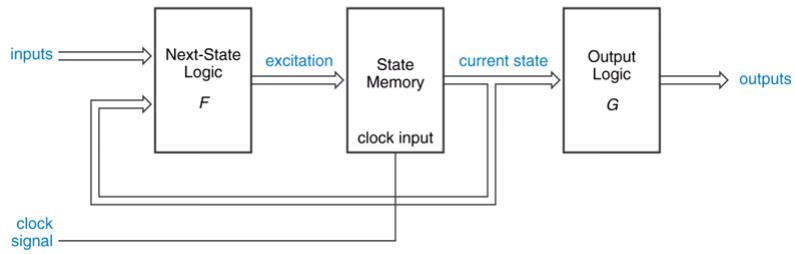
(Mealy Machine)

Next State = F (current state, input)

Output = G (current state, input)



# Structure of a State Machine

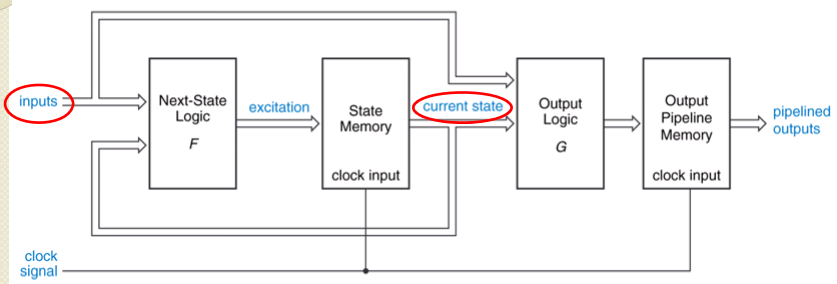



Clocked Synchronous State Machine

(Moore Machine)

Output = G (current state)

# Structure of a State Machine




Mealy Machine with Pipelined Outputs



# Instruction Pipeline

| Instr. No. | Pipeline Stage | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | IF | ID | EX | MEM | WB | | |
| 2 | | IF | ID | EX | MEM | WB | |
| 3 | | | IF | ID | EX | MEM | WB |
| 4 | | | | IF | ID | EX | MEM |
| 5 | | | | | IF | ID | EX |
| Clock Cycle | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

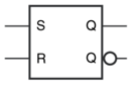
# Various Instruction Pipelines

| Micro-Architecture | Pipeline Stages |
|---|---|
| Intel P5 (Pentium) | 5 |
| Intel P6 (Pentium Pro) | 14 |
| Intel P6 (Pentium III) | 10 |
| IBM PowerPC 7 | 17 |
| IBM Xenon | 19 |
| AMD Athlon | 10 |
| AMD Athlon XP | 11 |
| AMD Athlon64 | 12 |
| AMD Phenom | 12 |
| AMD Opteron | 15 |



# Characteristic Equation

**Describing the function of a circuit:**

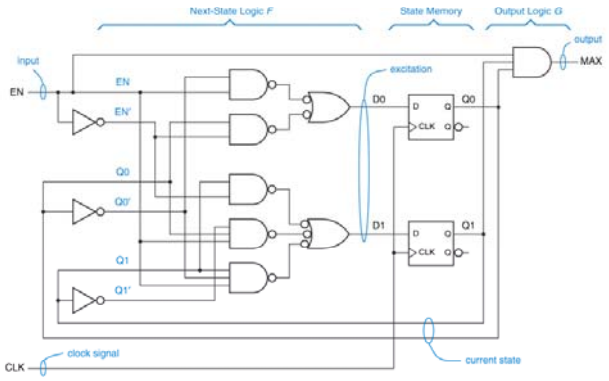



| S | R | Q | QN |
|---|---|---|---|
| 0 | 0 | last Q | last QN |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |

| *Device Type* | *Characteristic Equation* |
|---|---|
| S-R latch | $Q^* = S + R' \cdot Q$ |
| D latch | $Q^* = D$ |
| Edge-triggered D flip-flop | $Q^* = D$ |
| D flip-flop with enable | $Q^* = EN \cdot D + EN' \cdot Q$ |
| Master/slave S-R flip-flop | $Q^* = S + R' \cdot Q$ |
| Master/slave J-K flip-flop | $Q^* = J \cdot Q' + K' \cdot Q$ |
| Edge-triggered J-K flip-flop | $Q^* = J \cdot Q' + K' \cdot Q$ |
| T flip-flop | $Q^* = Q'$ |
| T flip-flop with enable | $Q^* = EN \cdot Q' + EN' \cdot Q$ |

# Analysis

Definition of a state machine:
Next state = F (current state, input)
Output = G (current state, input)






# Evaluation

(a)

| Q1 Q0 | EN = 0 | EN = 1 |
|---|---|---|
| 00 | 00 | 01 |
| 01 | 01 | 10 |
| 10 | 10 | 11 |
| 11 | 11 | 00 |
| | Q1* Q0* | |

(b)

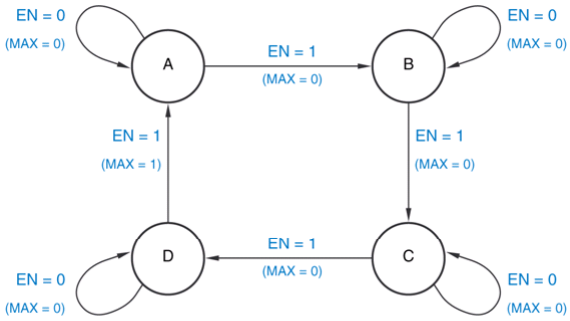
| S | EN = 0 | EN = 1 |
|---|---|---|
| A | A | B |
| B | B | C |
| C | C | D |
| D | D | A |
| | S* | |

(c)

| S | EN = 0 | EN = 1 |
|---|---|---|
| A | A, 0 | B, 0 |
| B | B, 0 | C, 0 |
| C | C, 0 | D, 0 |
| D | D, 0 | A, 1 |
| | S*, MAX | |

**D0 = Q0 EN' + Q0' EN**
**D1 = Q1 EN' + Q1' Q0 EN + Q1 Q0' EN**

# State Diagram

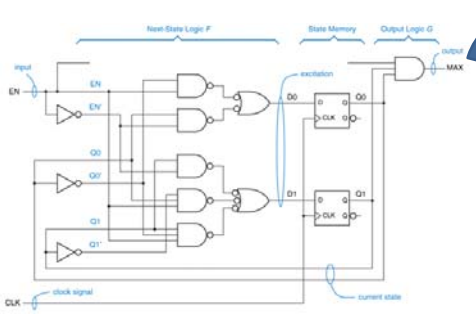



| | EN | |
|---|---|---|
| S | 0 | 1 |
| A | A, 0 | B, 0 |
| B | B, 0 | C, 0 |
| C | C, 0 | D, 0 |
| D | D, 0 | A, 1 |

S*, MAX



# State/Output Table for Moore Machine Model



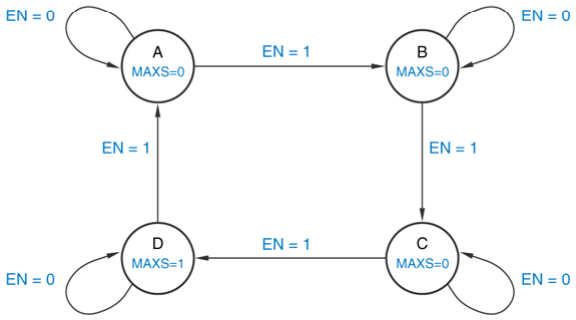
| | EN | | |
|---|---|---|---|
| S | 0 | 1 | MAXS |
| A | A | B | 0 |
| B | B | C | 0 |
| C | C | D | 0 |
| D | D | A | 1 |

S*

| | EN | |
|---|---|---|
| Q1 Q0 | 0 | 1 |
| 00 | 00 | 01 |
| 01 | 01 | 10 |
| 10 | 10 | 11 |
| 11 | 11 | 00 |

Q1* Q0*

# State Diagram




| S | EN = 0 | EN = 1 | MAXS |
|---|---|---|---|
| A | A | B | 0 |
| B | B | C | 0 |
| C | C | D | 0 |
| D | D | A | 1 |

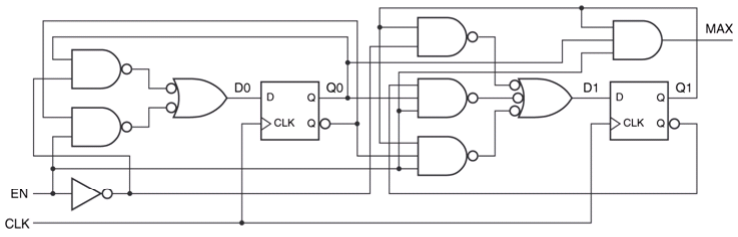
S*

| Q1 Q0 | EN = 0 | EN = 1 |
|---|---|---|
| 00 | 00 | 01 |
| 01 | 01 | 10 |
| 10 | 10 | 11 |
| 11 | 11 | 00 |

Q1* Q0*



# Modified/Redrawn Logic Diagram

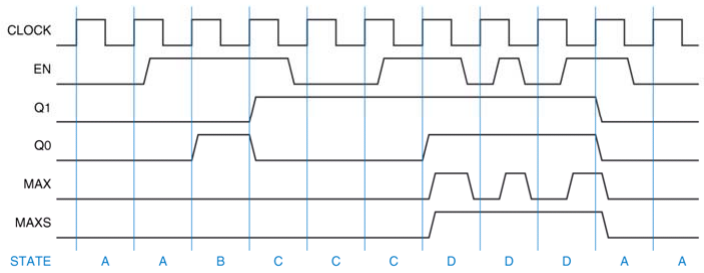


# Timing Diagram

| Q1 Q0 | EN = 0 | EN = 1 |
|---|---|---|
| 00 | 00 | 01 |
| 01 | 01 | 10 |
| 10 | 10 | 11 |
| 11 | 11 | 00 |
| | Q1+ Q0+ | |

| S | EN = 0 | EN = 1 |
|---|---|---|
| A | A, 0 | B, 0 |
| B | B, 0 | C, 0 |
| C | C, 0 | D, 0 |
| D | D, 0 | A, 1 |
| | S+, MAX | |

| S | EN = 0 | EN = 1 | MAXS |
|---|---|---|---|
| A | A | B | 0 |
| B | B | C | 0 |
| C | C | D | 0 |
| D | D | A | 1 |
| | S+ | | |

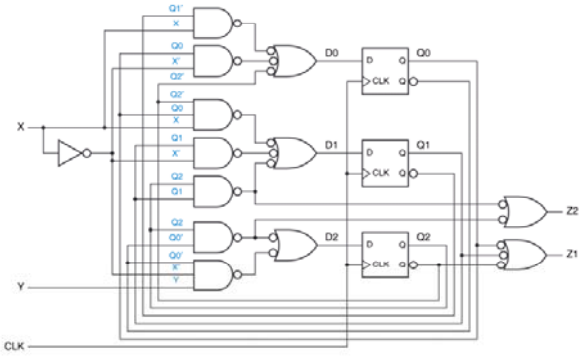





# Summary

1. Determine the excitation equations for the flip-flops.
2. Substitute them into the characteristics equations to get the transition equations.
3. Use the transition equations to construct a transition table.
4. Determine the output equations.
5. Create a transition/output table.
6. Name the states.
7. Draw a state diagram.

## State Machine with 3 Flip-flops & 8 states






## Transition/Output & State/Output Table

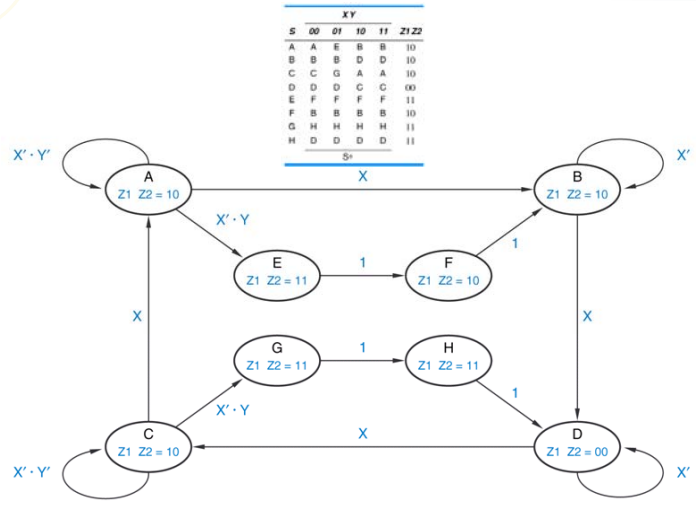
(a)

| Q2 Q1 Q0 | XY = 00 | 01 | 10 | 11 | Z1 Z2 |
|---|---|---|---|---|---|
| 000 | 000 | 100 | 001 | 001 | 10 |
| 001 | 001 | 001 | 011 | 011 | 10 |
| 010 | 010 | 110 | 000 | 000 | 10 |
| 011 | 011 | 011 | 010 | 010 | 00 |
| 100 | 101 | 101 | 101 | 101 | 11 |
| 101 | 001 | 001 | 001 | 001 | 10 |
| 110 | 111 | 111 | 111 | 111 | 11 |
| 111 | 011 | 011 | 011 | 011 | 11 |
| | Q2* Q1* Q0* | | | | |

(b)

| S | XY = 00 | 01 | 10 | 11 | Z1 Z2 |
|---|---|---|---|---|---|
| A | A | E | B | B | 10 |
| B | B | B | D | D | 10 |
| C | C | G | A | A | 10 |
| D | D | D | C | C | 00 |
| E | F | F | F | F | 11 |
| F | B | B | B | B | 10 |
| G | H | H | H | H | 11 |
| H | D | D | D | D | 11 |
| | S* | | | | |

# State Diagram





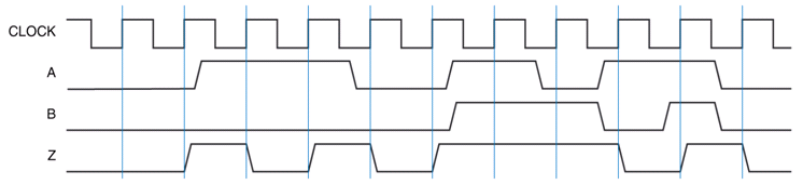

# Designing a Clocked Synchronous State Machine

1. Construct a state / Output table.
2. If possible minimise the number of states.
3. Choose a set of state variable for state assignments.
4. Create a transition/output table by substituting state variable combinations into state/output table.
5. Select / Choose a flip flop, almost always it is D type flip flop.
6. Construct an excitation table, which shows the excitation values required to get to the next desired state for all possible state/input combinations.
7. Derive excitation equations from excitation table.
8. Derive output equations from the transition table.
9. Draw a logic diagram.

# State Table Design

**Design a clocked synchronous state machine with two inputs A, B and output Z so that Z is in state 1 if:**

**A had the same value at each of the two previous clock ticks or**
**B has been 1 since the last time the first condition was met/true**
**otherwise, output should be 0.**






# Design Steps

1. From this verbal description create/construct a state / Output table.

2. Minimise the number of states. The basic idea is that two or more states are equivalent if it is impossible to distinguish by observing the current and future output of the machine. Typically, these techniques are rarely used, because some time increasing the number of states simplifies the design, most engineers simplify the machine during state assignment (next step).

| | AB | | | | |
|---|---|---|---|---|---|
| Q1 Q2 Q3 | 00 | 01 | 11 | 10 | Z |
| 000 | 100 | 100 | 101 | 101 | 0 |
| 100 | 110 | 110 | 101 | 101 | 0 |
| 101 | 100 | 100 | 111 | 111 | 0 |
| 110 | 110 | 110 | 111 | 101 | 1 |
| 111 | 100 | 110 | 111 | 111 | 1 |
| | Q1* | Q2* | Q3* | | |

| | AB | | | | |
|---|---|---|---|---|---|
| Q1Q2Q3 | 00 | 01 | 11 | 10 | Z |
| A | B | B | C | C | 0 |
| B | D | D | C | C | 0 |
| C | B | B | E | E | 0 |
| D | D | D | E | C | 1 |
| E | B | D | E | E | 1 |

# Possible State Assignments

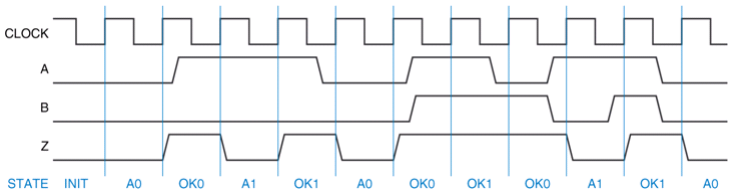
| State Name | Assignment | | | |
|---|---|---|---|---|
| | Simplest Q1–Q3 | Decomposed Q1–Q3 | One-Hot Q1–Q5 | Almost One-Hot Q1–Q4 |
| INIT | 000 | 000 | 00001 | 0000 |
| A0 | 001 | 100 | 00010 | 0001 |
| A1 | 010 | 101 | 00100 | 0010 |
| OK0 | 011 | 110 | 01000 | 0100 |
| OK1 | 100 | 111 | 10000 | 1000 |



# Disposition of States

- **Minimal Risk**
- **Minimal Cost**

# Timing Diagram & State Sequence






# Synthesis using D Type Flip-Flop

| | AB | | | | |
|---|---|---|---|---|---|
| Q1 Q2 Q3 | 00 | 01 | 11 | 10 | Z |
| 000 | 100 | 100 | 101 | 101 | 0 |
| 100 | 110 | 110 | 101 | 101 | 0 |
| 101 | 100 | 100 | 111 | 111 | 0 |
| 110 | 110 | 110 | 111 | 101 | 1 |
| 111 | 100 | 110 | 111 | 111 | 1 |
| | Q1* Q2* Q3* | | | | |

**Transition & Output Table**

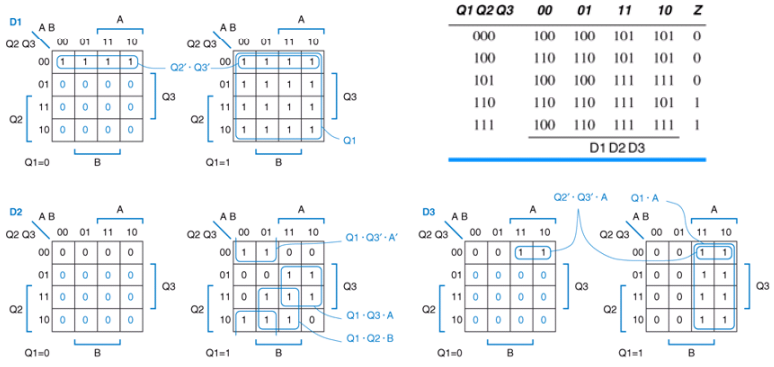
# State Machine

| Q1 Q2 Q3 | AB 00 | 01 | 11 | 10 | Z |
|---|---|---|---|---|---|
| 000 | 100 | 100 | 101 | 101 | 0 |
| 100 | 110 | 110 | 101 | 101 | 0 |
| 101 | 100 | 100 | 111 | 111 | 0 |
| 110 | 110 | 110 | 111 | 101 | 1 |
| 111 | 100 | 110 | 111 | 111 | 1 |
| | D1 D2 D3 | | | | |

**Excitation & Output Table**



# Excitation Maps




| Q1 Q2 Q3 | AB 00 | 01 | 11 | 10 | Z |
|---|---|---|---|---|---|
| 000 | 100 | 100 | 101 | 101 | 0 |
| 100 | 110 | 110 | 101 | 101 | 0 |
| 101 | 100 | 100 | 111 | 111 | 0 |
| 110 | 110 | 110 | 111 | 101 | 1 |
| 111 | 100 | 110 | 111 | 111 | 1 |
| | D1 D2 D3 | | | | |

D1 = Q2' . Q3' + Q1
D2 = Q1 . Q3' . A' + Q1 . Q3 . A + Q1 . Q2 . B
D3 = Q1 . A + Q2' . Q3' . A
Z = Q1 . Q2 . Q3' + Q1 Q2 Q3
Z = Q1 . Q2

# 1s-Counting Machine

**Design a clocked synchronous state machine with 2 inputs X, Y and output Z. The output should be 1 if the number of inputs on X and Y since reset is a multiple of 4 and 0 otherwise.**

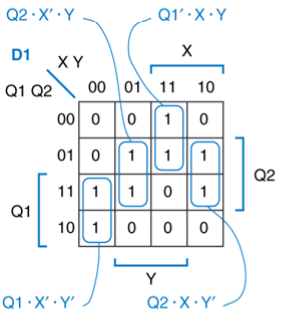
| | | XY | | | | |
|---|---|---|---|---|---|---|
| *Meaning* | *S* | *00* | *01* | *11* | *10* | *Z* |
| Got zero 1s (modulo 4) | S0 | S0 | S1 | S2 | S1 | 1 |
| Got one 1 (modulo 4) | S1 | S1 | S2 | S3 | S2 | 0 |
| Got two 1s (modulo 4) | S2 | S2 | S3 | S0 | S3 | 0 |
| Got three 1s (modulo 4) | S3 | S3 | S0 | S1 | S0 | 0 |
| | | S* | | | | |

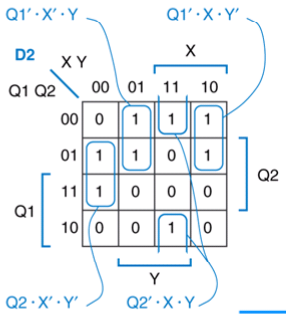


# Transition/Excitation & Output Table

| | XY | | | | |
|---|---|---|---|---|---|
| *Q1 Q2* | *00* | *01* | *11* | *10* | *Z* |
| 00 | 00 | 01 | 11 | 01 | 1 |
| 01 | 01 | 11 | 10 | 11 | 0 |
| 11 | 11 | 10 | 00 | 10 | 0 |
| 10 | 10 | 00 | 01 | 00 | 0 |
| | Q1* Q2* or D1 D2 | | | | |

# Excitation Maps




| | | XY | | | |
|---|---|---|---|---|---|
| Q1 Q2 | 00 | 01 | 11 | 10 | Z |
| 00 | 00 | 01 | 11 | 01 | 1 |
| 01 | 01 | 11 | 10 | 11 | 0 |
| 11 | 11 | 10 | 00 | 10 | 0 |
| 10 | 10 | 00 | 01 | 00 | 0 |
| | Q1* Q2* or D1 D2 | | | | |

D1 = Q2 . X' . Y + Q1' . X . Y + Q1 . X' . Y' + Q2 . X . Y'

D2 = Q1' . X' . Y + Q1' . X . Y' + Q2 . X' . Y' + Q2' . X . Y

Z = Q1' . Q2'



# State & Output Table for Combination-lock Machine

| | | X | |
|---|---|---|---|
| Meaning | S | 0 | 1 |
| Got zip | A | B, 01 | A, 00 |
| Got 0 | B | B, 00 | C, 01 |
| Got 01 | C | B, 00 | D, 01 |
| Got 011 | D | E, 01 | A, 00 |
| Got 0110 | E | B, 00 | F, 01 |
| Got 01101 | F | B, 00 | G, 01 |
| Got 011011 | G | E, 00 | H, 01 |
| Got 0110111 | H | B, 11 | A, 00 |
| | | S*, UNLK HINT | |

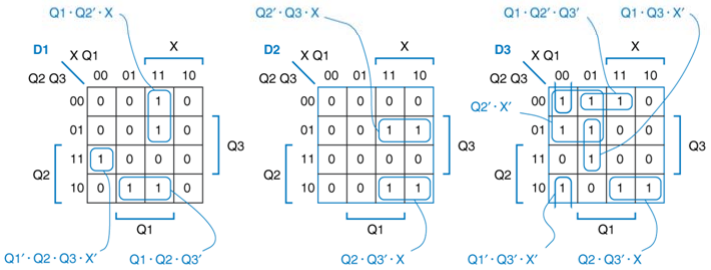
# Transition/Excitation Table for Combination-lock Machine

| | X | |
|---|---|---|
| Q1 Q2 Q3 | 0 | 1 |
| 000 | 001, 01 | 000, 00 |
| 001 | 001, 00 | 010, 01 |
| 010 | 001, 00 | 011, 01 |
| 011 | 100, 01 | 000, 00 |
| 100 | 001, 00 | 101, 01 |
| 101 | 001, 00 | 110, 01 |
| 110 | 100, 00 | 111, 01 |
| 111 | 001, 11 | 000, 00 |

Q1* Q2* Q3*, UNLK HINT



# Excitation Map

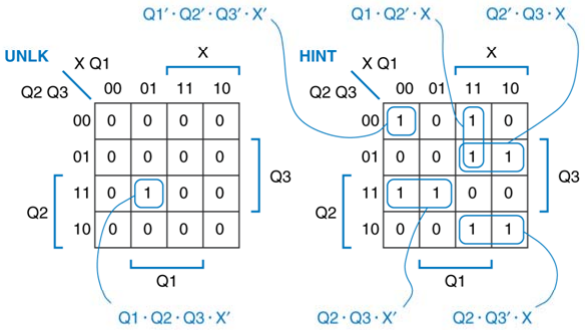



D1 = Q1 . Q2' . X + Q1' . Q2 . Q3 + Q1 . Q2 . Q3'

D2 = Q2' . Q3 . X + Q2 . Q3' . X

D3 = Q1 . Q2' . Q3' + Q1 . Q3 . X' + Q2' . X' . + Q3' . Q1' . X' + Q2 . Q3' . X

# Karnaugh Maps




UNLK = Q1 . Q2 . Q3 . X'

HINT = Q1' . Q2' . Q3' . X' + Q1 . Q2' . X + Q2' . Q3 . X + Q2 . Q3 . X' + Q2 . Q3' . X



# Spørgsmål
# Opgaver