**Tugas Mandiri - 6**

Pengantar Sistem Digital

Semester Ganjil 2022/2023

Petunjuk pengerjaan:

* Kerjakan dengan tulisan tangan atau diketik.
* Tuliskan Nama, Kelas, dan NPM pada setiap lembar jawaban.
* Tuliskan penjelasan dari cara mendapatkan jawaban tersebut.
* Apabila ditulis tangan, hasil pekerjaan di scan / foto dan dimasukan ke dalam satu file berformat .pdf.
* Format nama file (tanpa tanda kurung) : **[KodeAsdos]\_TM6\_[Nama]\_[NPM].pdf** dan **[KodeAsdos]\_TM6\_[Nama]\_[NPM].circ (dikumpulkan 2 file)**.
* Tugas mandiri dikumpulkan Jumat, 18 November 2022 pukul 17.00 pada slot yang sudah disediakan di SCELE.
* Jika **mengumpulkan telat di atas 10 menit tetapi sebelum pukul 23:59 pada hari yang sama**, akan dikenakan **penalti sebesar 50 poin**. Terlebih dari waktu tersebut, tugas mandiri **tidak akan dinilai**

1. (10 poin) Buatlah state table dari fungsi berikut:

Buatlah dua state table: state table satu dimensi dan state table dua dimensi.

**State Table Dua Dimensi** (Moore)

| **Present State** | | **Next State** | | **Output** | |
| --- | --- | --- | --- | --- | --- |
| **x(t) = 0** | **x(t) = 1** | **x(t) = 0** | **x(t) = 1** |
| **A(t)** | **B(t)** | **A(t+1)B(t+1)** | **A(t+1)B(t+1)** | **Y(t)** | **Y(t)** |
| 0 | 0 | 01 | 01 | 0 | 1 |
| 0 | 1 | 11 | 00 | 0 | 1 |
| 1 | 0 | 01 | 00 | 0 | 1 |
| 1 | 1 | 01 | 11 | 0 | 0 |

**State Table Satu Dimensi** (Mealy)

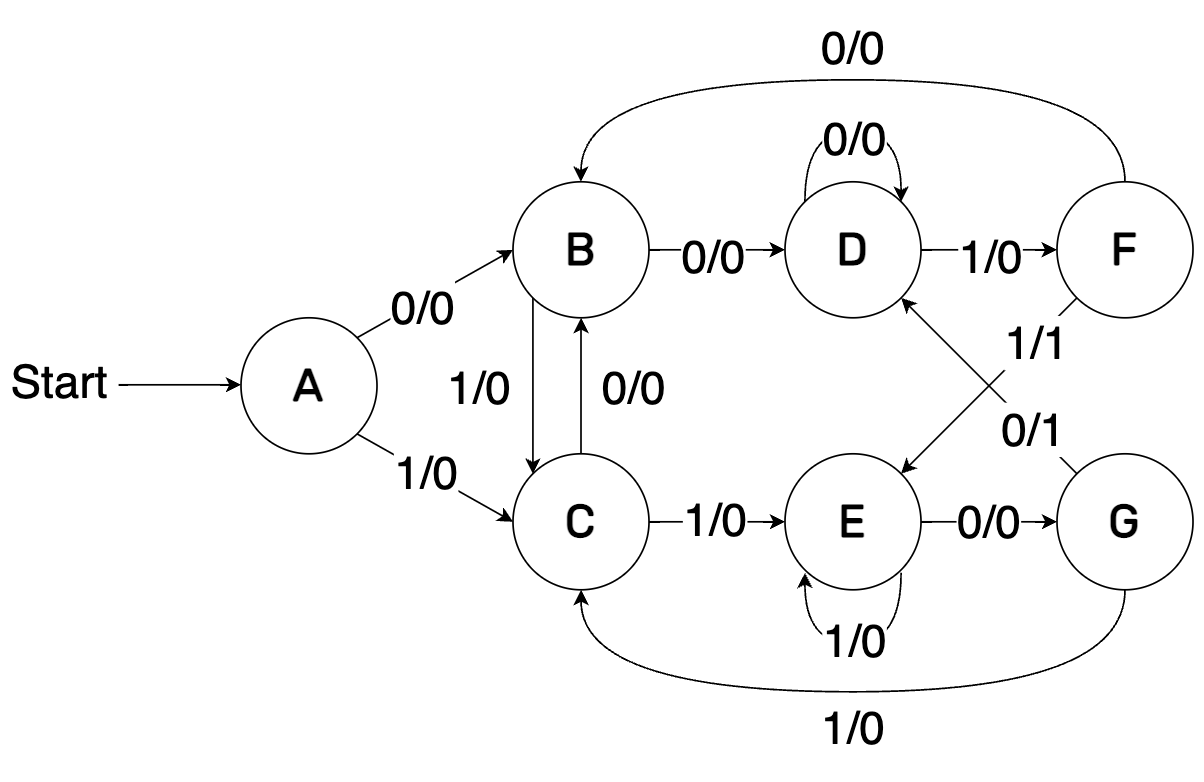
| **Present State** | | **Input** | **Next State** | | **Output** |
| --- | --- | --- | --- | --- | --- |
| **A(t)** | **B(t)** | **x(t)** | **A(t+1)** | **B(t+1)** | **Y(t)** |
| 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 |

**Untuk soal nomor 2 dan 3, kumpulkan juga circuitnya dalam satu file .circ yang sama.**

1. (50 poin) Diketahui ada sebuah sequential circuit yang menerima 1 buah input dan 1 buah output. Circuit tersebut merupakan sequence recognizer yang dapat mengenali input “0011” dan “1100”. Bila circuit menerima salah satu sequence input tersebut, maka circuit akan menghasilkan output 1 (selain itu outputnya 0).
2. Buatlah state diagramnya (Mealy model)
3. Buatlah state tablenya
4. Buatlah gambar rangkaiannya dengan menggunakan gerbang AND, OR, NOT dan D Flip Flop saja. Tuliskan langkah-langkah design procedurenya (kecuali tahap verification)

Catatan: untuk tahap state assignment, gunakan counting order assignment.

1. **State Diagram**



1. **State Table**

| **Present State** | **Input (X)** | **Next State** | **Output (Y)** |
| --- | --- | --- | --- |
| A | 0 | B | 0 |
| A | 1 | C | 0 |
| B | 0 | D | 0 |
| B | 1 | C | 0 |
| C | 0 | B | 0 |
| C | 1 | E | 0 |
| D | 0 | D | 0 |
| D | 1 | F | 0 |
| E | 0 | G | 0 |
| E | 1 | E | 0 |
| F | 0 | B | 0 |
| F | 1 | E | 1 |
| G | 0 | D | 1 |
| G | 1 | C | 0 |

1. **Design Procedure**

**Specification:**

Membuat sebuah sequence recognizer yang dapat mengenali “0011” dan “1100”.

Output akan bernilai 1 jika menerima salah satu sequence input tersebut.

State menggunakan 3 bit binary.

Input = X

Output = Y

**Formulation:**

State Diagram dan State Table terlampir di bagian 2a) dan 2b).

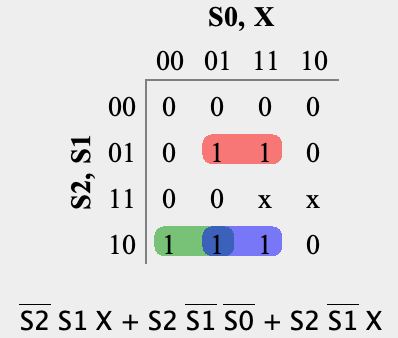
**State Assignment:**

| **State** | **Encoding (S2 : S0)** |
| --- | --- |
| A | 000 |
| B | 001 |
| C | 010 |
| D | 011 |
| E | 100 |
| F | 101 |
| G | 110 |

| **Present State** | | | **Input** | **Next State** | | | **Output** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **S2** | **S1** | **S0** | **X** | **S2+** | **S1+** | **S0+** | **Y** |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | x | x | x | x |
| 1 | 1 | 1 | 1 | x | x | x | x |

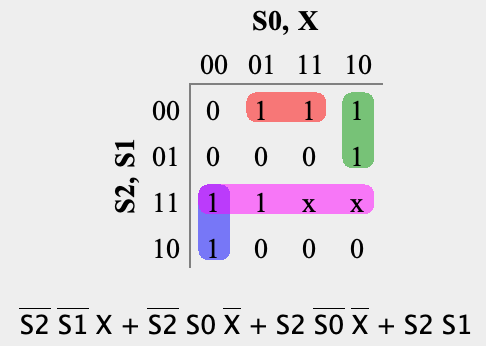
**Persamaan Flip-Flop dan Output serta Optimization**

S2+:



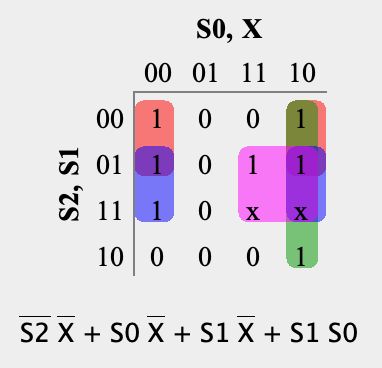
S2+ = S2’ S1 X + S2 S1’ S0’ + S2 S1’ X

S1+:



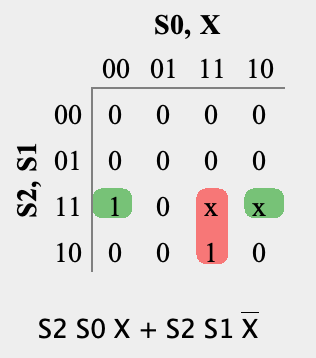
S1+ = S2’ S1’ X + S2’ S0 X’ + S2 S0’ X’ + S2 S1

S0+:



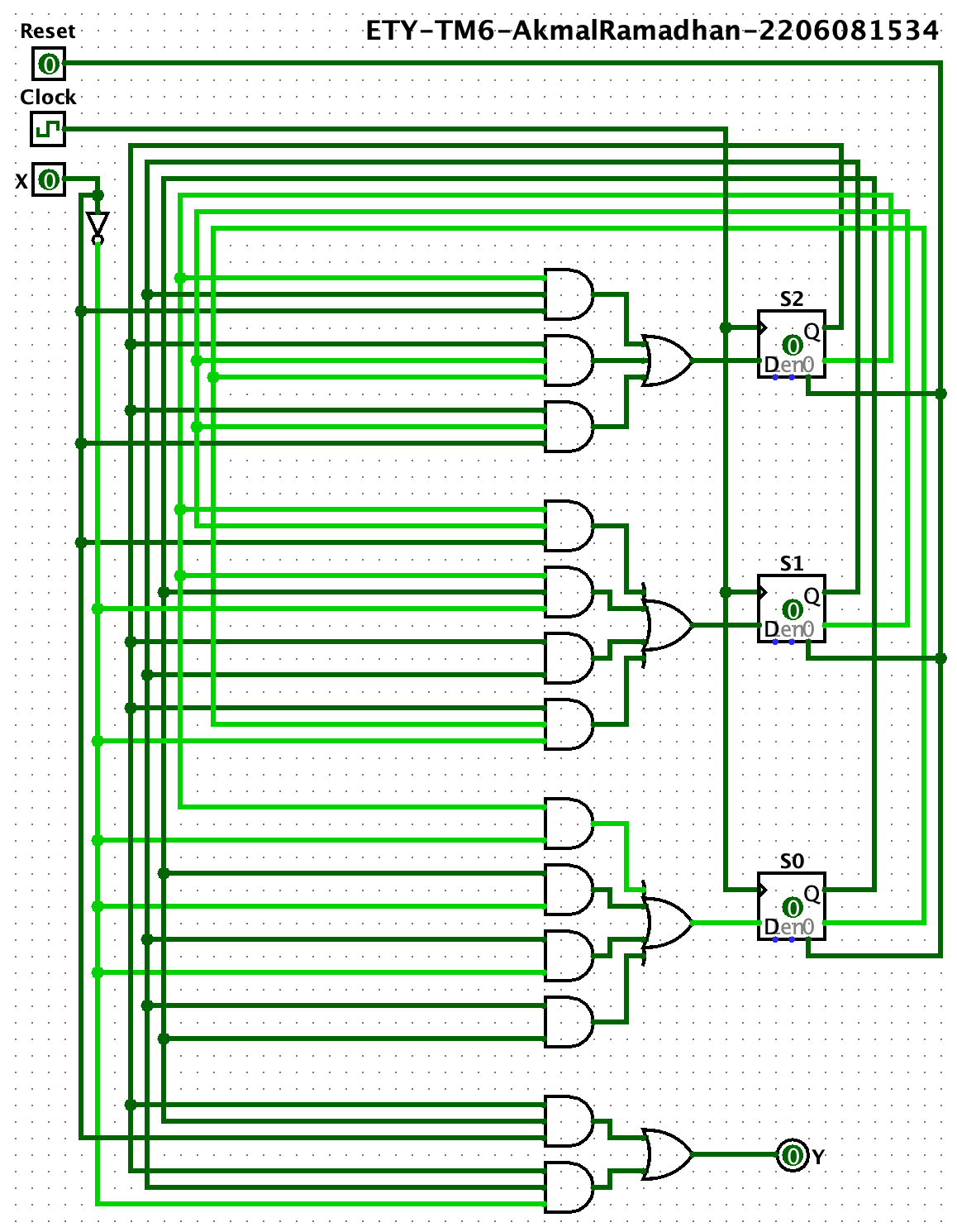
S0+ = S2’ X’ + S0 X’ + S1 X’ + S1 S0

Y:



Y = S2 S0 X + S2 S1 X’

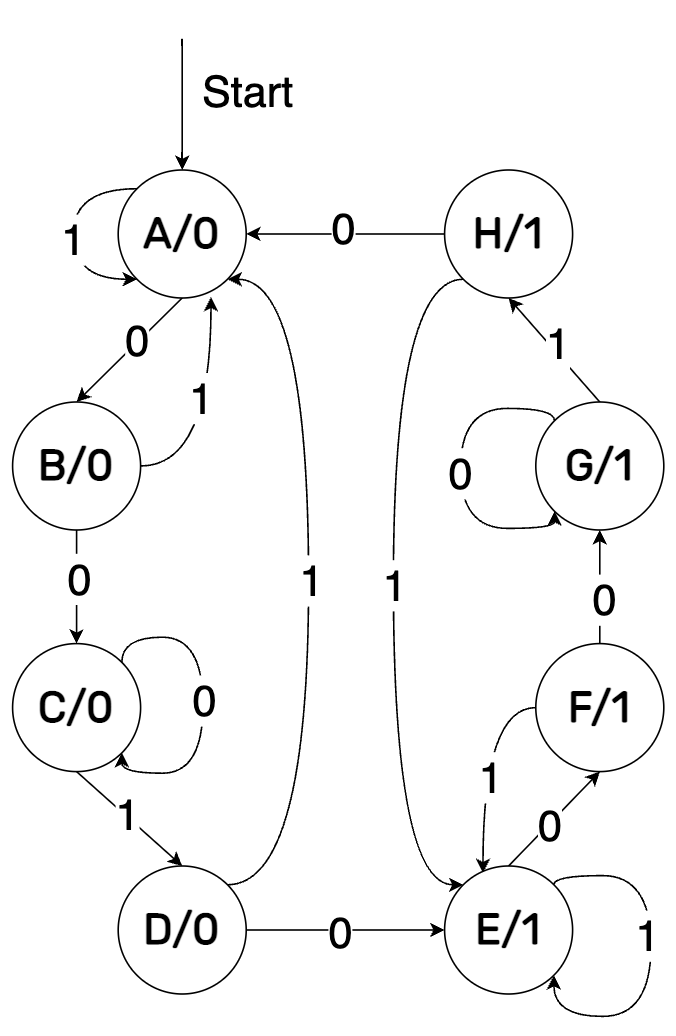
**Technology Mapping**

****

1. (40 poin) Diketahui ada sebuah sequential circuit yang menerima 1 buah input dan 1 buah output. Circuit tersebut merupakan sequence recognizer yang dapat mengenali input yang mengandung substring “0010” non-overlap berjumlah ganjil (contoh: 0010, 001011, 001000100010, dan sebagainya). Bila circuit menerima input tersebut, maka circuit akan menghasilkan output 1 (selain itu outputnya 0).
2. Buatlah state diagramnya (Moore model)
3. Buatlah state tablenya
4. Buatlah gambar rangkaiannya dengan menggunakan gerbang AND, OR, NOT dan JK Flip Flop saja. Tuliskan langkah-langkah design procedurenya (kecuali tahap verification)

Catatan: untuk tahap state assignment, gunakan counting order assignment.

1. **State Diagram**



1. **State Table**

| **Present State** | **Input (X)** | **Next State** | **Output (Y)** |
| --- | --- | --- | --- |
| A | 0 | B | 0 |
| A | 1 | A | 0 |
| B | 0 | C | 0 |
| B | 1 | A | 0 |
| C | 0 | C | 0 |
| C | 1 | D | 0 |
| D | 0 | E | 1 |
| D | 1 | A | 0 |
| E | 0 | F | 1 |
| E | 1 | E | 1 |
| F | 0 | G | 1 |
| F | 1 | E | 1 |
| G | 0 | G | 1 |
| G | 1 | H | 1 |
| H | 0 | A | 0 |
| H | 1 | E | 1 |

1. **Design Procedure**

**Specification:**

Membuat sebuah sequence recognizer yang dapat mengenali input yang mengandung substring “0010” non-overlap berjumlah ganjil.

Bila circuit menerima input tersebut, maka circuit akan menghasilkan output 1 (selain itu outputnya 0).

Input = X

Output = Y

**Formulation:**

State Diagram dan State Table terlampir di bagian 2a) dan 2b).

**State Assignment:**

| **State** | **Encoding (S2 : S0)** |
| --- | --- |
| A | 000 |
| B | 001 |
| C | 010 |
| D | 011 |
| E | 100 |
| F | 101 |
| G | 110 |
| H | 111 |

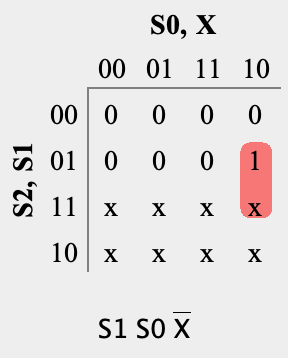
| **Present State** | **Next State** | | **JK Flip-Flop** | | | | | | **Output** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **x(t) = 0** | **x(t) = 1** | **x(t) = 0** | | | **x(t) = 1** | | |
| **S2(t) S1(t) S0(t)** | **S2(t+1) S1(t+1) S0(+1)** | **S2(t + 1) S1(t+1) S0(t+1)** | **JS2 KS2** | **JS1 KS1** | **JS0 KS0** | **JS2 KS2** | **JS1 KS1** | **JS0 KS0** | **Y(t)** |
| 0 0 0 | 0 0 1 | 0 0 0 | 0 x | 0 x | 1 x | 0 x | 0 x | 0 x | 0 |
| 0 0 1 | 0 1 0 | 0 0 0 | 0 x | 1 x | x 1 | 0 x | 0 x | x 1 | 0 |
| 0 1 0 | 0 1 0 | 0 1 1 | 0 x | x 0 | 0 x | 0 x | x 0 | 1 x | 0 |
| 0 1 1 | 1 0 0 | 0 0 0 | 1 x | x 1 | x 1 | 0 x | x 1 | x 1 | 0 |
| 1 0 0 | 1 0 1 | 1 0 0 | x 0 | 0 x | 1 x | x 0 | 0 x | 0 x | 1 |
| 1 0 1 | 1 1 0 | 1 0 0 | x 0 | 1 x | x 1 | x 0 | 0 x | x 1 | 1 |
| 1 1 0 | 1 1 0 | 1 1 1 | x 0 | x 0 | 0 x | x 0 | x 0 | 1 x | 1 |
| 1 1 1 | 0 0 0 | 1 0 0 | x 1 | x 1 | x 1 | x 0 | x 1 | x 1 | 1 |

Mealy Version (Extra):

| **Present State** | | | **Input** | **Next State** | | | **JK** | | | | | | **Output** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **S2** | **S1** | **S0** | **X** | **S2+** | **S1+** | **S0+** | **JS2** | **KS2** | **JS1** | **KS1** | **JS0** | **KS0** | **Y** |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | x | 0 | x | 1 | x | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | x | 0 | x | 0 | x | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | x | 1 | x | x | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | x | 0 | x | x | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | x | x | 0 | 0 | x | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | x | x | 0 | 1 | x | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | x | x | 1 | x | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | x | x | 1 | x | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | x | 0 | 0 | x | 1 | x | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | x | 0 | 0 | x | 0 | x | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | x | 0 | 1 | x | x | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 | 0 | x | 0 | 0 | x | x | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | x | 0 | x | 0 | 0 | x | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | x | 0 | x | 0 | 1 | x | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | x | 1 | x | 1 | x | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | x | 0 | x | 1 | x | 1 | 1 |

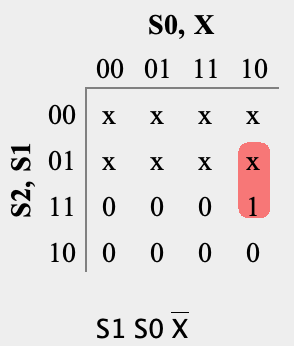
**Persamaan Flip-Flop dan Output serta Optimization**

JS2:



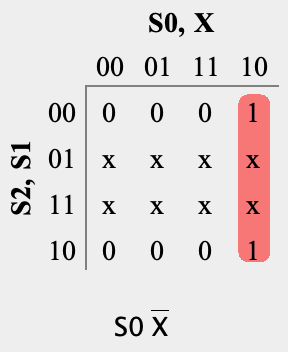
JS2 = S1 S0 X’

KS2:



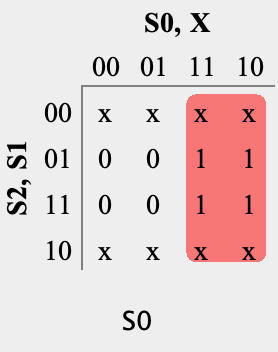
KS2 = S1 S0 X’

JS1:



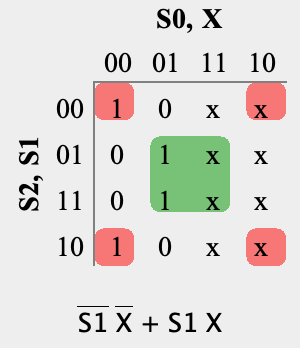
JS1 = S0 X’

KS1:



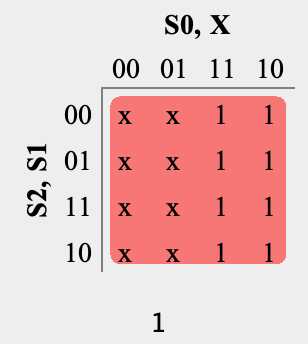
KS1 = S0

JS0:



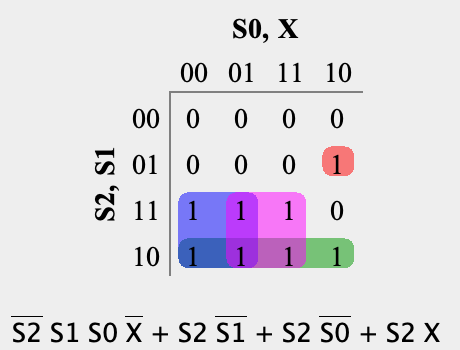
JS0 = S1’ X’ + S1 X

KS0:



KS0 = 1

Y:



Y = S2’ S1 S0 X’ + S2 S1’ + S2 S0’ + S2 X

**Technology Mapping**

