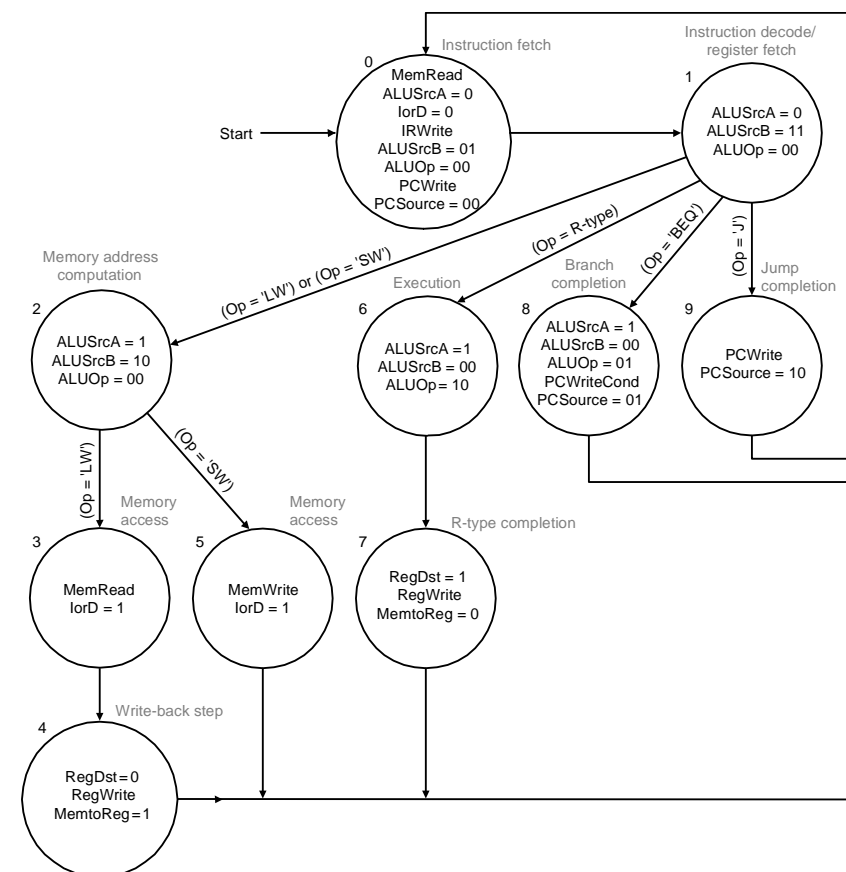
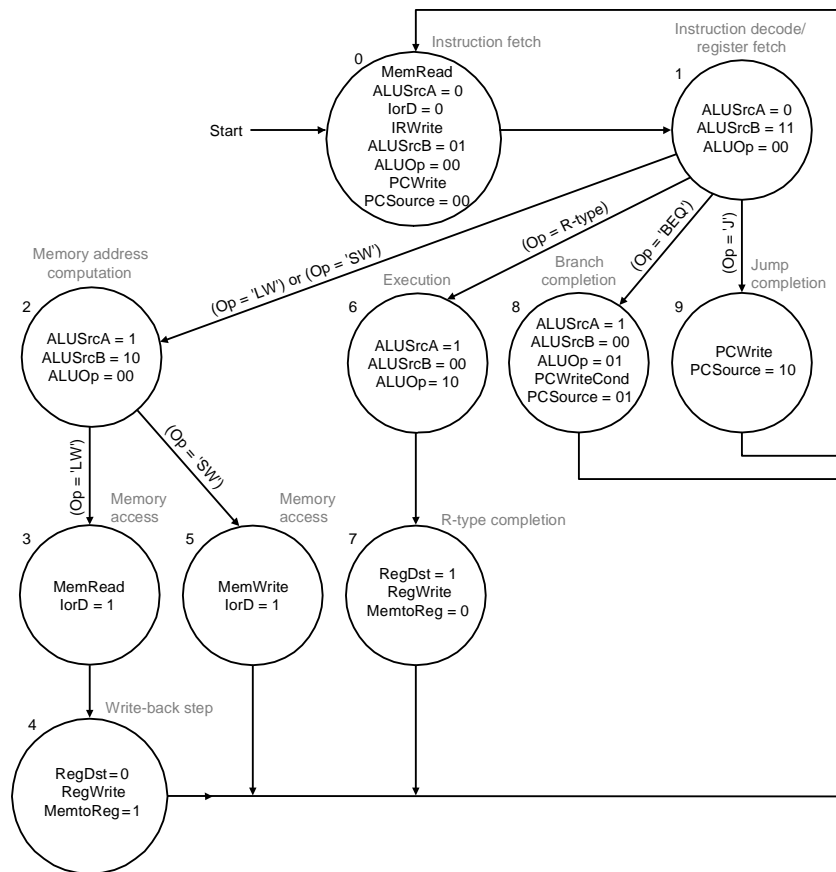


EE 044252: Digital Systems and Computer Structure

Spring 2018

Lecture 6: More *Finite State Machines*

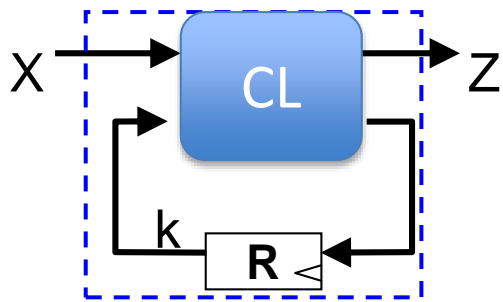


EE 044252: Digital Systems and Computer Structure

Topic	wk	Lectures	Tutorials	Workshop	Simulation
Arch	1	Intro. RISC-V architecture	Numbers. Codes		
Comb	2	Switching algebra & functions	Assembly programming		
	3	Combinational logic	Logic minimization	Combinational	
	4	Arithmetic. Memory	Gates		Combinational
Seq	5	Finite state machines	Logic		
	6	Sync FSM	Flip flops, FSM timing	Sequential	Sequential
	7	FSM equiv, scan, pipeline	FSM synthesis		
	8	Serial comm, memory instructions	Serial comm, pipeline		
μArch	9	Function call, single cycle RISC-V	Function call		
	10	Multi-cycle RISC-V	Single cycle RISC-V		Multi-cycle
	11	Interrupts, pipeline RISC-V	Multi-cycle RISC-V		
	12	Dependencies in pipeline RISC-V	Microcode, interrupts		
	13		Depend. in pipeline RISC-V		

Agenda

- FSM timing
- Clock skew
- FSM limitations



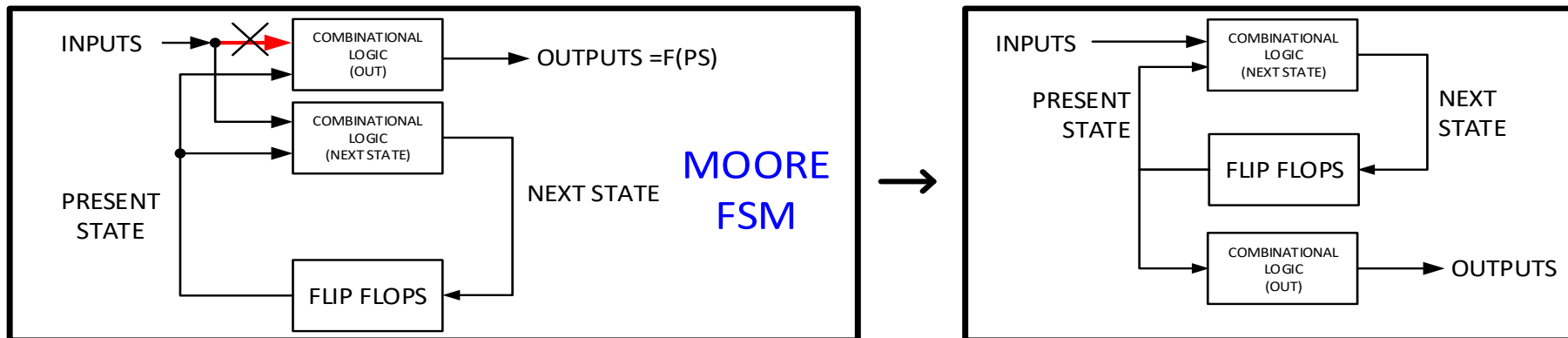
הגדרת FSM

- מערכת עקיבה ממומשת ע"י מכונת מצבים סופית (Finite State-Machine, FSM) המוגדרת באמצעות מרכיביה
 - קבוצה סופית של מצבים $S = \{s_1, s_2, \dots, s_K\}$. אחד מהם נקבע כמצב ההתחלתי. ניתן לייצג K מצבים ע"י $k = \lceil \log_2(K) \rceil$ משתני מצב בינאריים
 - קבוצה סופית של כניסות בינאריות $X = \{x_1, x_2, \dots, x_L\}$
 - קבוצה סופית של יציאות בינאריות $Z = \{z_1, z_2, \dots, z_M\}$
 - פונקציית מעבר $\lambda(S \times 2^X \rightarrow S)$ המגדירה לכל צירוף של מצב נוכחי s_i וערכי הכניסות x_1, x_2, \dots, x_L את המצב הבא s_i^*
 - פונקציית יציאה $\Omega_{\text{MEALY}}(S \times 2^X \rightarrow 2^Z)$ המגדירה לכל צירוף של מצב נוכחי s_i וערכי הכניסות x_1, x_2, \dots, x_L את ערכי היציאות z_1, z_2, \dots, z_M
 - תזמוני כניסה t_H , t_S ותזמוני יציאה $t_{\text{PC-Q}}$, $t_{\text{cC-Q}}$
- מכונת מצבים כזו קרויה ע"ש **Mealy**
- לעומתה, במכונת **Moore** שונה פונקציית היציאה והיא תלויה **במצב הנוכחי בלבד** :

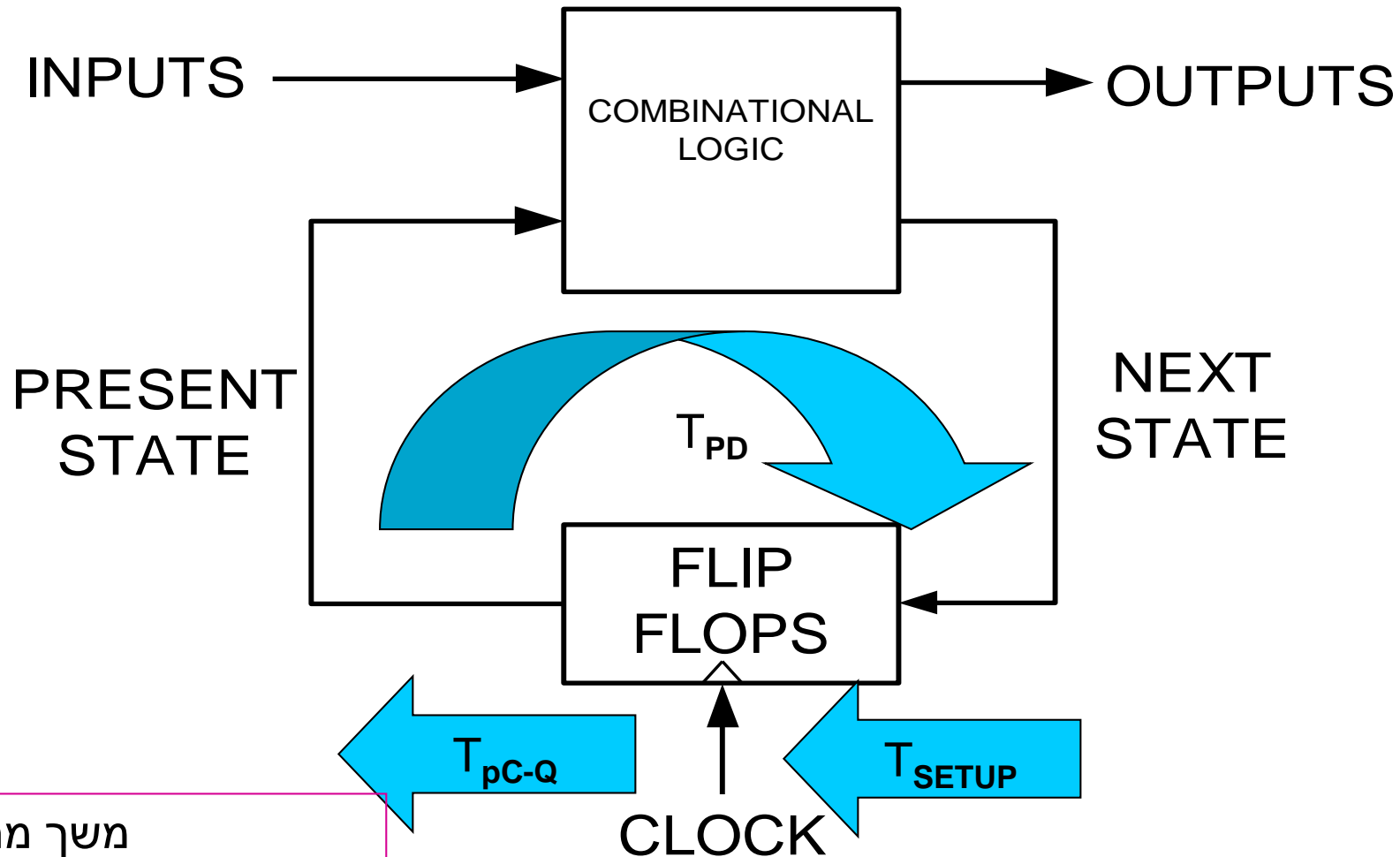
$$\Omega_{\text{MOORE}}(S \rightarrow 2^Z)$$

מכונת מצבים – Moore או Mealy?

- Mealy לתיאור תיאורטי
 - המכונה מוגבלת לגילוי מלים השייכות לשפה
 - לא נשתמש לבניית מערכות ספרתיות המורכבות ממספר מכונות
 - הסיבה תובהר בהמשך (קושי בהגבלת האורך של מסלולים צירופיים)
- Moore לבניית מערכות מורכבות
 - נמנע מסלול צירופי מכניסה ליציאה



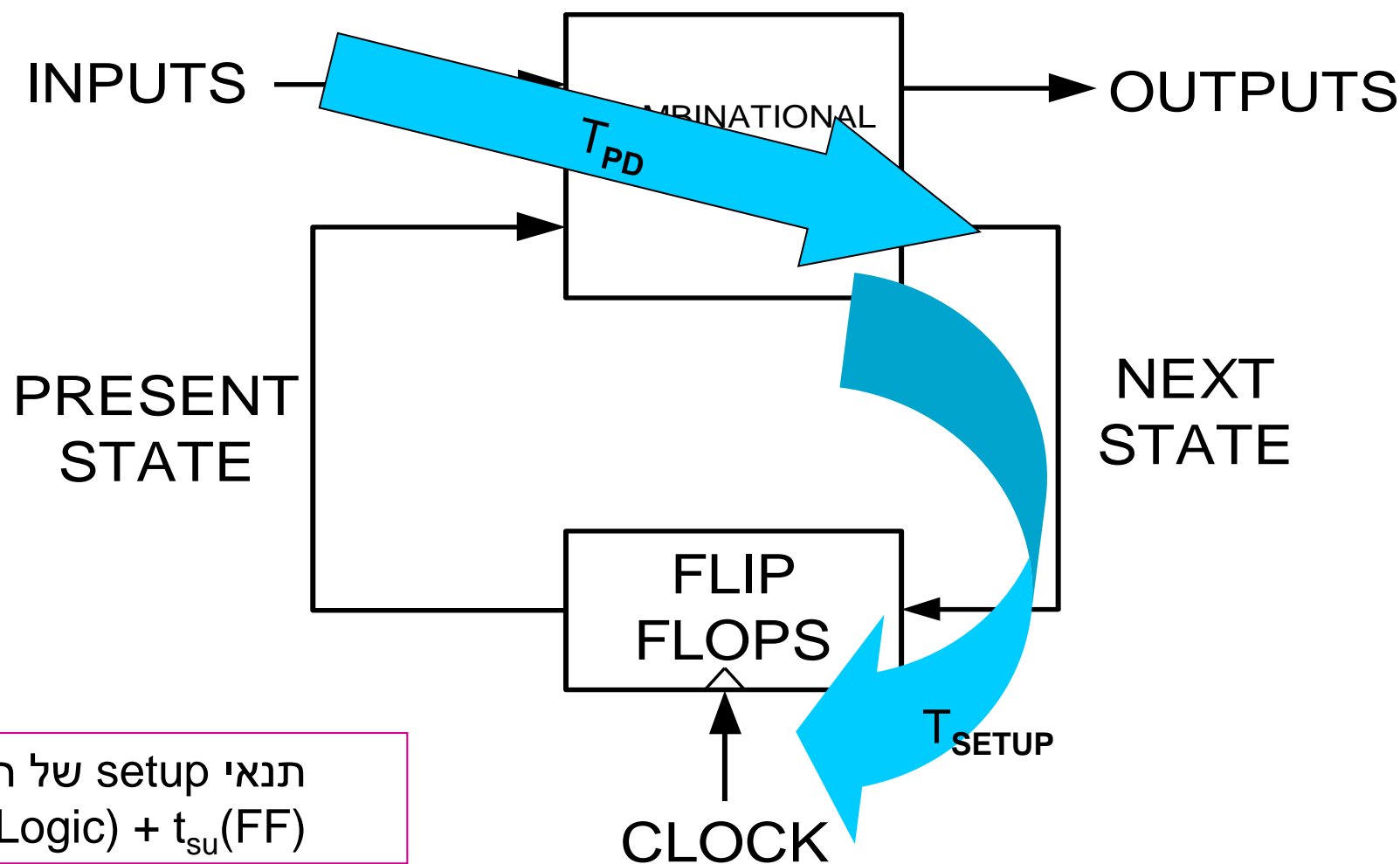
תזמון במכונת מצבים : זמן המחזור



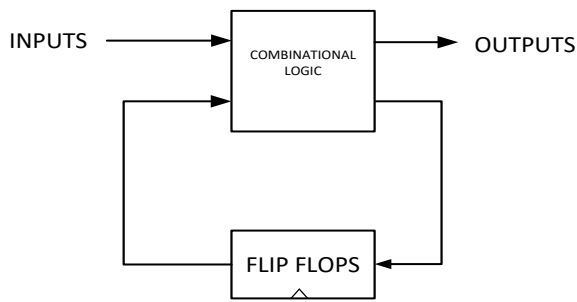
משך מחזור השעון T :

$$T \geq t_{pCQ}(FF) + t_{pd}(C.Logic) + t_{su}(FF)$$

תזמון במכונת מצבים : זמן SETUP



תנאי setup של הכניסה למכונה:
 $t_{su}(IN) \geq t_{pd}(C.Logic) + t_{su}(FF)$



כללי התזמון

- ישנם ארבעה מסלולים, שניים שראינו: שעות \rightarrow שעות, קלט \rightarrow קלט, ושניים נוספים: פלט \rightarrow קלט, פלט \rightarrow שעות
- בכדי להבטיח פעולה תקינה של מערכת עקיבה יש להקפיד על שני כללים לכל מסלול. נרשום חלק מהם:

– משך מחזור השעות T :

$$T \geq t_{pC-Q} + t_{pd}(C.Logic) + t_s$$

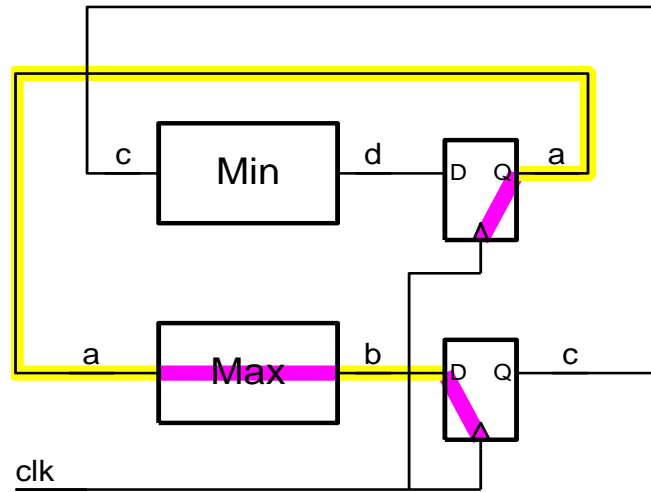
– הכניסות למערכת הצירופית צריכות להיות תקפות בערכים הנכונים במשך $t_s(input)$:

$$t_s(input) \geq t_{pd}(C.Logic) + t_s$$

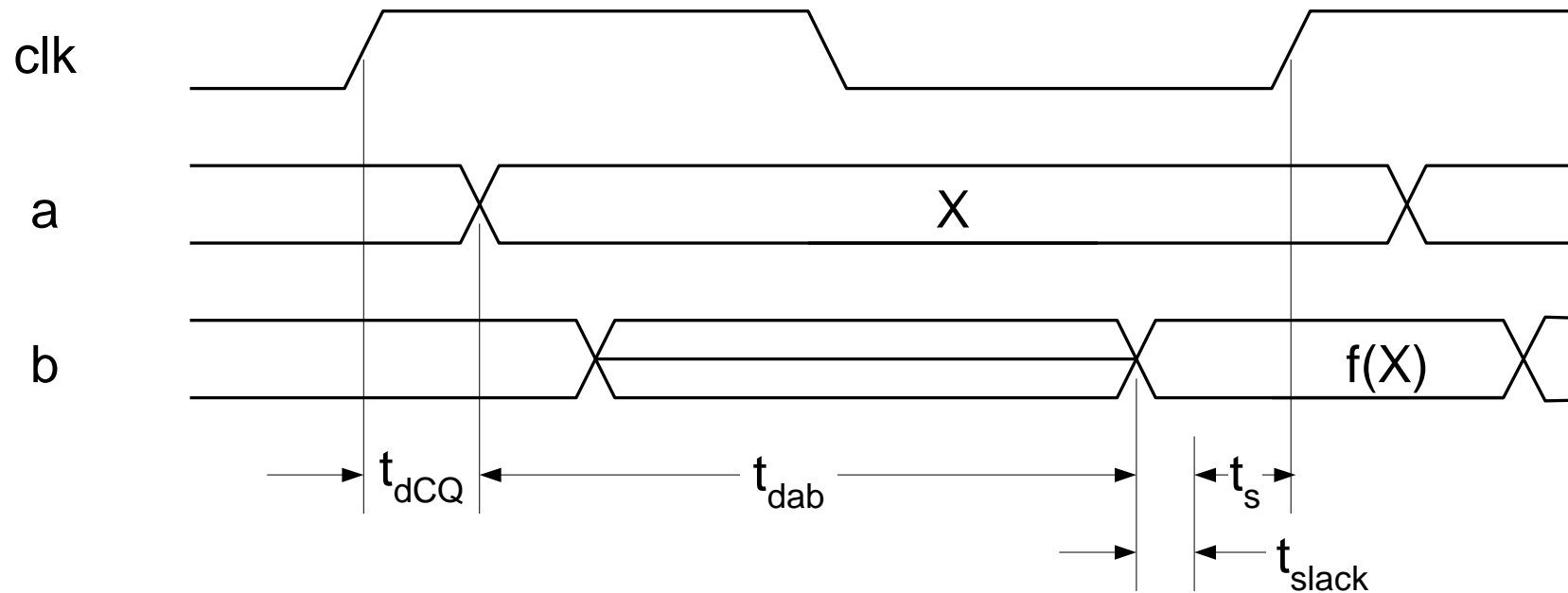
– $t_{cd}(C.Logic) + t_{c-Q}$ במערכת הצירופית צריך להיות ארוך מאשר t_H של הזיכרונות

– הכניסות למערכת הצירופית צריכות להיות תקפות בערכים הנכונים במשך $t_H(input)$ המקיים: $t_H(input) \geq t_H - t_{cd}(C.Logic)$

Setup Time Constraint



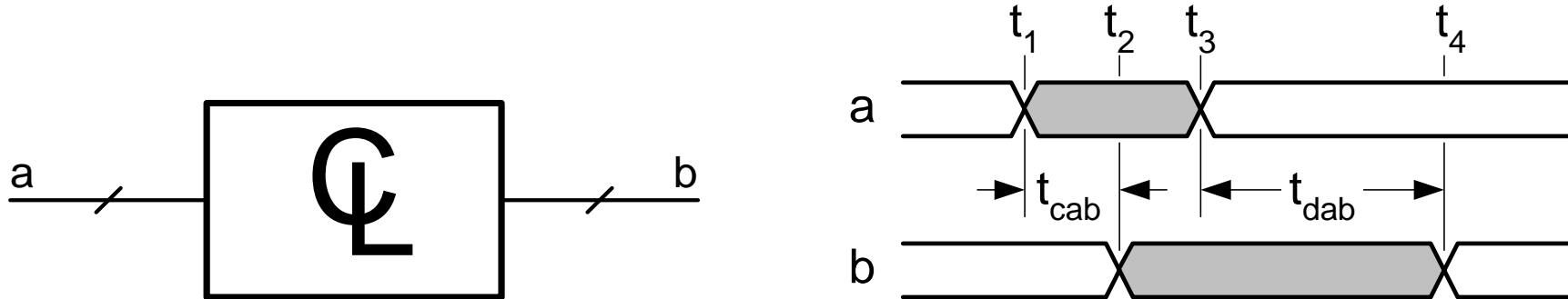
$$t_{cy} > t_{dCQ} + t_{dMax} + t_s$$



Propagation Delay and Contamination Delay

Propagation Delay – Time from last input change until last output change. (Input at steady state to output at steady state.)

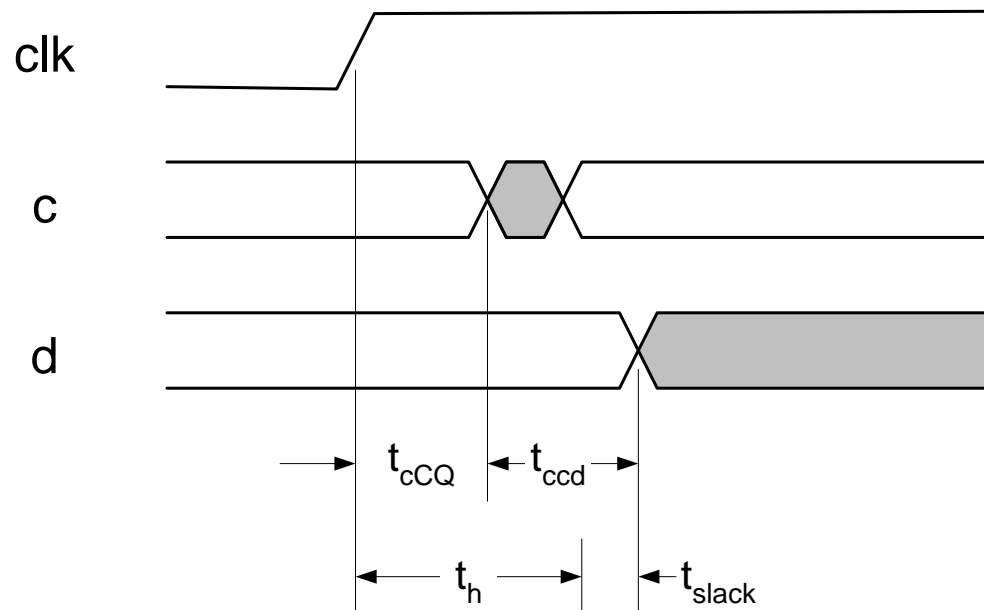
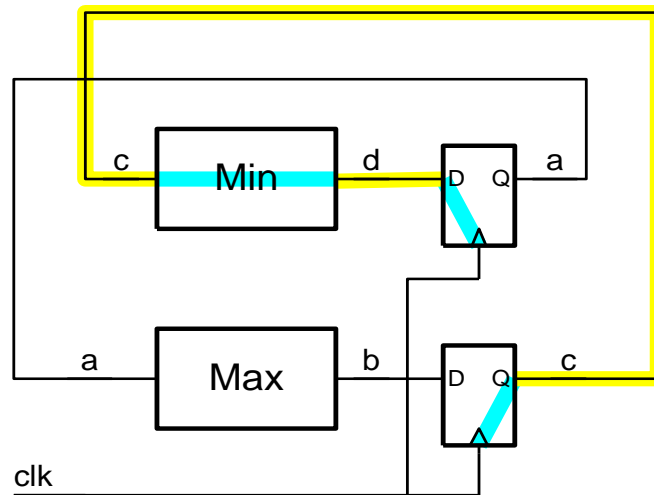
Contamination Delay – Time from first input change until first output change. (Input contaminated to output contaminated)



Hold Time Constraint

$$t_h < t_{cCQ} + t_{cMin}$$

Unsafe at any speed



t_{cXY} – contamination delay
 t_{dXY} – propagation delay

Example

$$t_{dcQ} = t_{ccQ} = t_s = 150\text{ps}$$

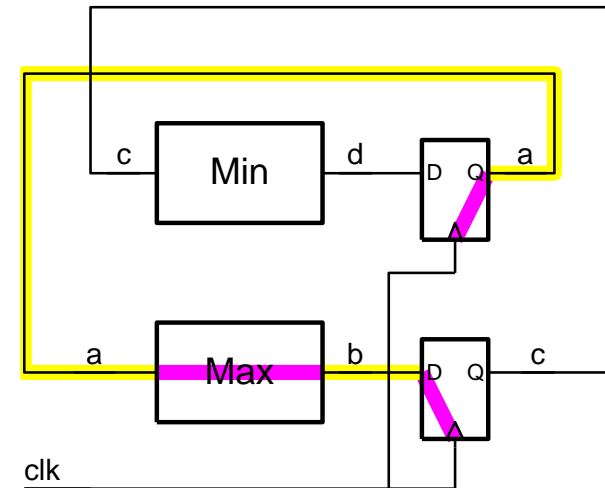
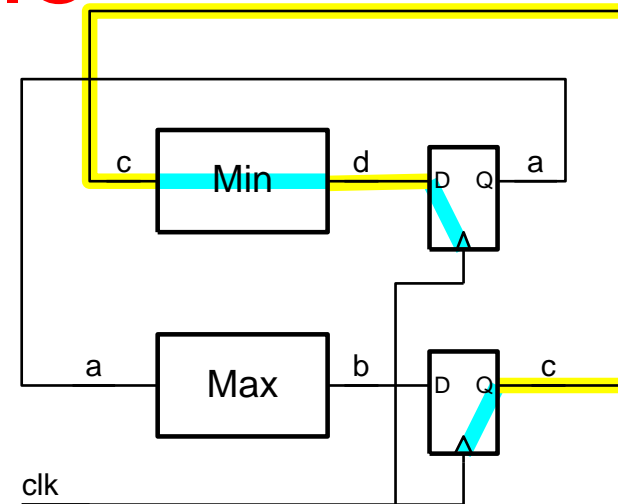
$$t_h = 250\text{ps}$$

$$t_{dMax} = 850\text{ps}$$

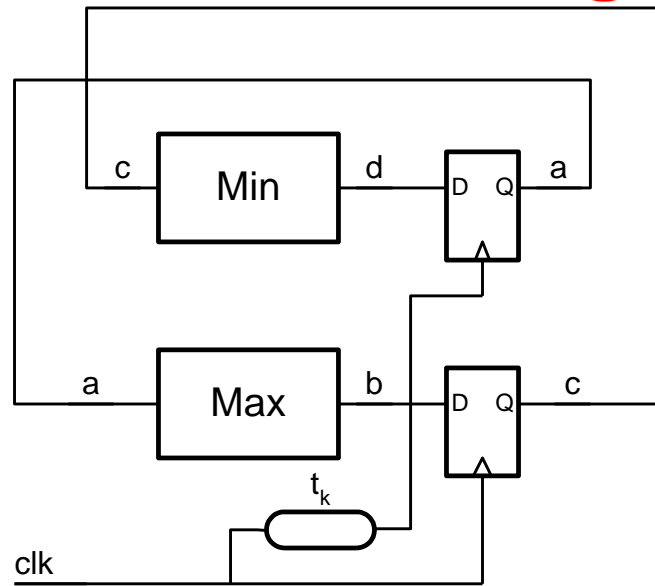
$$t_{cMin} = 100\text{ps}$$

Is hold time constraint met?

What is the minimum cycle time?

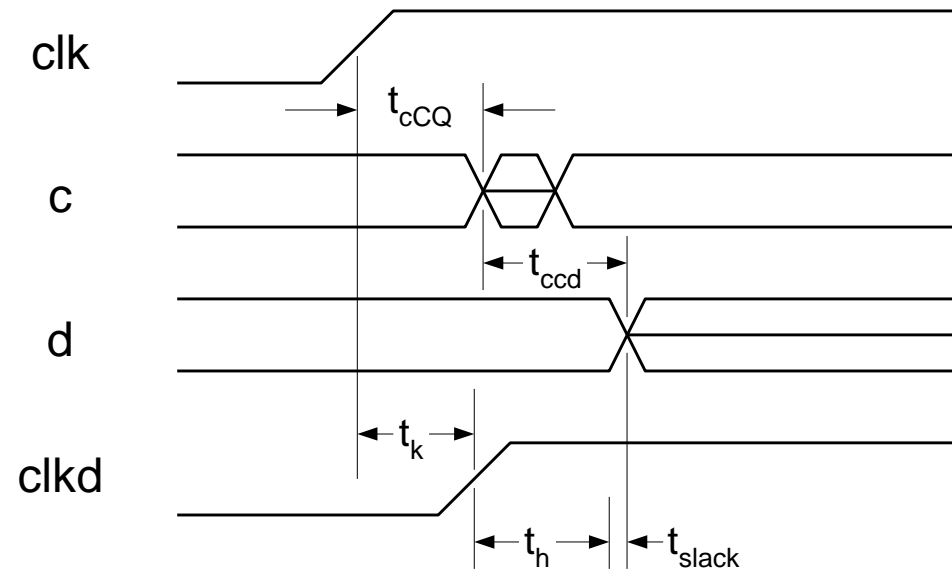


Clock Skew

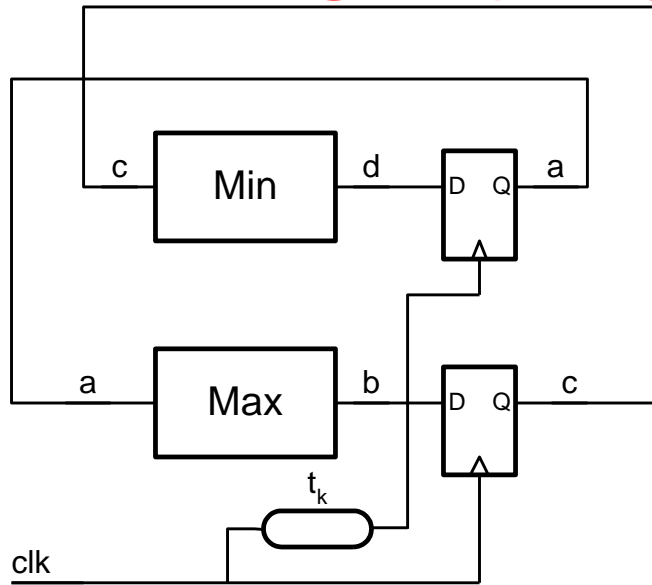


$$t_{cy} > t_{dCQ} + t_{dMax} + t_s + t_k$$

$$t_h < t_{cCQ} + t_{cMin} - t_k$$



Clock Skew--Example



$$t_{cy} > t_{dCQ} + t_{dMax} + t_s + t_k$$

$$t_h < t_{cCQ} + t_{cMin} - t_k$$

$$t_{dCQ} = t_{cCQ} = t_s = 150\text{ps}$$

$$t_h = 250\text{ps}$$

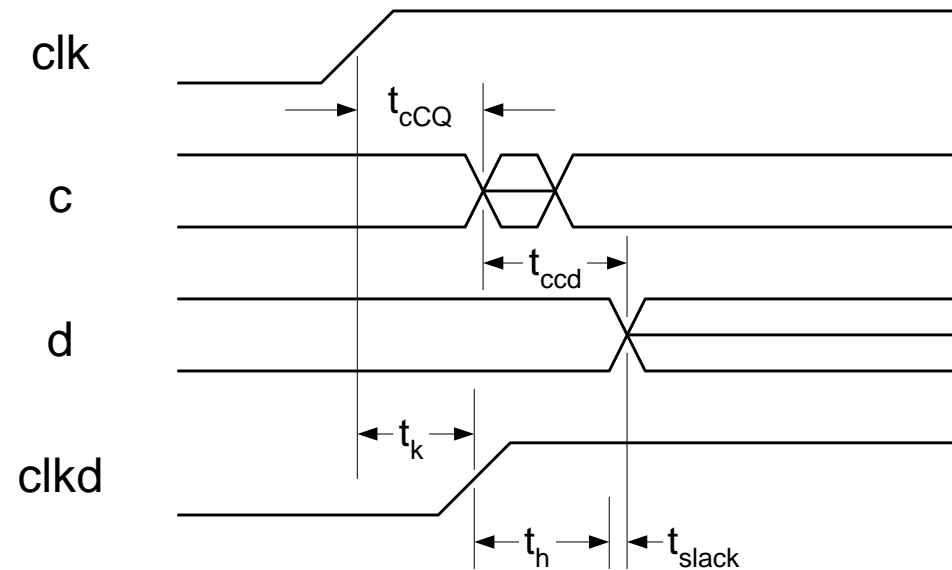
$$t_{dMax} = 850\text{ps}$$

$$t_{cMin} = 100\text{ps}$$

$$t_k = 100\text{ps}$$

Is hold time constraint met?

What is the minimum cycle time?



There is also jitter...

שלבי התכנון של מערכת עקיבה

1. למד את התיאור המילולי של המערכת הנדרשת
 - הבנת הדרישות : מספר כניסות, יציאות, מצבי זיכרון
2. בנה טבלת מצבים או דיאגרמת מצבים
3. (צמצם את טבלת המצבים — נלמד בהמשך)
4. בחר הקצאת מצבים ובחר רכיבי זיכרון
 - מסוג D-FF
5. רשום את טבלת המעברים ואת טבלת היציאה
6. מצא את פונקציות המעבר ופונקציות היציאה
7. שרטט את המעגל המממש את המערכת

תכונות ומגבלות של מערכות עקיבה

- מערכות העקיבה מומשו עד כה באמצעות זיכרון סופי, ולכן מספר המצבים הוא סופי. מכונות כאלו קרויות Finite State Machine (FSM)
- האם ניתן לחשב באמצעות מכונה כזו כל חישוב שהוא? האם ניתן לממש באמצעותה כל טרנספורמציה רצויה על הקלט?
- התשובה כמובן שלילית: מספר המצבים הסופי מטיל מגבלה על יכולת החישוב
- נניח כי למכונה n מצבים, והקלט הוא סדרה של 1 -ים, ארוכה מ- n . עם קריאת הקלט המכונה תעבור על סדרת מצבים

$$S_{t(0)} \xrightarrow{1} S_{t(1)} \xrightarrow{1} S_{t(2)} \xrightarrow{1} \dots \xrightarrow{1} S_{t(i)} \xrightarrow{1} \dots \xrightarrow{1} S_{t(j)} \dots \xrightarrow{1} S_{t(n-1)} \xrightarrow{1} S_{t(n)} \xrightarrow{1} \dots$$

- נקבל, לראשונה, $S_{t(i)} = S_{t(j)}$ עבור $0 \leq i < j \leq n$, כלומר המכונה "שכחה" שהיה הבדל ביניהם
- מובן שהמכונה תמשיך מ- $S_{t(i)}$ ומ- $S_{t(j)}$ באותה סדרת מצבים (הרי הקלט הוא אותו קלט בשני המקרים: 1 -ים)
- באותו אופן, המכונה גם תייצר אותו פלט בשני המקרים
- במילים אחרות, אנו מקבלים פלט מחזורי!

דוגמה : מכונה להשוואת אורכי מחרוזות

- נתכנן מכונה המוציאה $Z=1$ אם"ם מספר ה-1ים שווה למספר האפסים. למשל:

$$X = 1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 1$$

$$Z = 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0$$

- ננסה לצייר דיאגרמת מצבים

דוגמה : מכונה להשוואת אורכי מחרוזות

- נתכנן מכונה המוציאה $Z=1$ אם"ם מספר ה-1ים שווה למספר האפסים. למשל:

$$X = 1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 1$$

$$Z = 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0$$

- נניח שקיימת מכונה עם n מצבים המבצעת את הנדרש. נספק לה את הקלט המכיל $n+1$ פעמים הספרה 1 ואחר כך $n+1$ פעמים הספרה 0.
- על המכונה להוציא 1 לראשונה רק עם תום קריאת הקלט. אולם, כיוון שאורך סדרת האפסים הוא $n+1$, המכונה נכנסה למחזור, ולכן תוציא 0!

משפט המחזוריות של מכונת מצבים סופית

- **משפט**: מכונה בעלת מספר מצבים סופי מפיקה יציאה מחזורית תחת קלט מחזורי, למעט "זמן-מעבר" סופי
- **הוכחה**: נניח שלמכונה n מצבים ואורך המחזור של סדרת הקלט הוא k סיביות. נבדוק מהו מצב המכונה בכל פעם שמחזור הקלט מתחיל מחדש (כלומר בכל k סיביות קלט). לאחר $n+1$ מחזורי קלט לכל היותר, לפחות שני מצבים יהיו זהים:

$$S_{t(0)} \xrightarrow{1} S_{t(1)} \xrightarrow{1} S_{t(2)} \xrightarrow{1} \dots \xrightarrow{1} S_{t(i)} \xrightarrow{1} \dots \xrightarrow{1} S_{t(j)} \dots \xrightarrow{1} S_{t(n-1)} \xrightarrow{1} S_{t(n)} \xrightarrow{1} \dots$$

- מאחר ששני המצבים זהים וסדרת הקלט זהה, גם סדרת הפלט תהייה זהה. לכן גם סדרת הפלט מחזורית. מ.ש.ל.
- מסקנה: חישובים מסוימים אינם ניתנים למימוש באמצעות מכונה סופית.
– למשל השוואת אורכי מחרוזות (אורך המחזור היה 1).

מכונה לזיהוי ריבועים

- נתכנן מכונה המוציאה $z=1$ אם מספר ה-1-ים בכניסה הוא ריבוע שלם (1, 4, 9, 16, ...)
- המכונה צריכה להוציא את הסדרה הבאה כאשר נתונה סדרת-קלט המכילה רק את הספרה 1:

X= 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1...

Z= 1 0 0 1 0 0 0 0 1 0 0 0 0 0 0 1...

- סדרת הפלט איננה מחזורית: עבור כל מספר n , המועמד להיות מחזור, קיים $k > 0$ כך שעבור $k \geq k_0$,

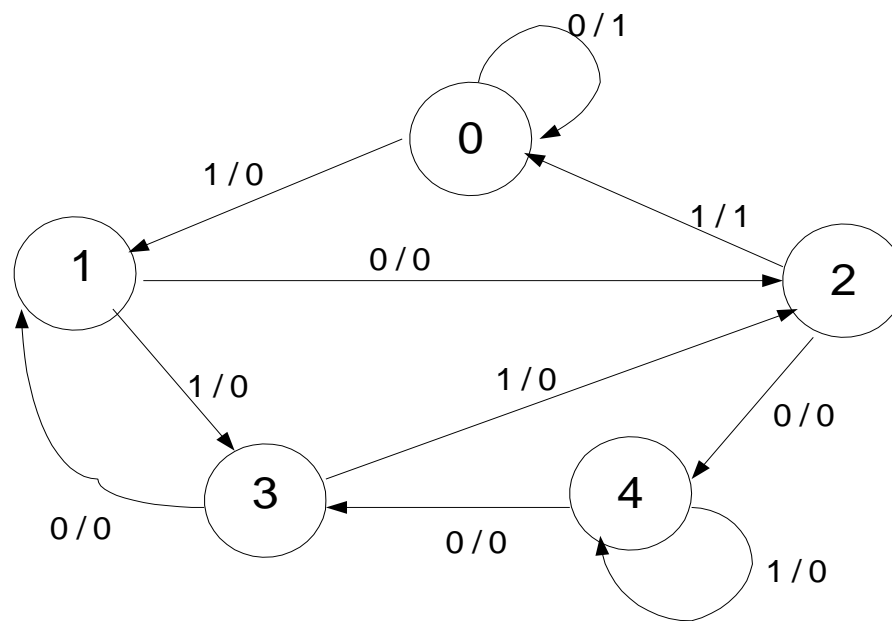
$$k^2 - (k - 1)^2 > n$$

- כלומר יהיו רצפים של יותר מ- n אפסים, למרות שהסדרה איננה זהותית 0!
- מסקנה: המכונה איננה ניתנת למימוש במספר סופי של מצבים

מכונה לחישוב השארית

- נתכנן מכונה המוציאה $z=1$ אם"ם המספר הבינרי המתקבל באופן טורי בקלט מתחלק ב-5 ללא שארית (ה- msb נכנס ראשון)
 - יש לזכור רק את השארית בחלוקה ב-5 של המספר שנקרא באופן טורי עד כה
- נניח שהמספר שהתקבל עד כה הוא x והשארית היא 0 , וכעת מגיעה הסיבית הבאה
 - המספר החדש שווה בערכו ל- $2x$ (כי x הוזז שמאלה בסיבית אחת) ועוד 0 (אם הסיבית החדשה היא 0) או ועוד 1 (אם הסיבית החדשה היא 1)
 - השארית החדשה היא 0 או 1 , בהתאמה.
- במקרה הראשון היציאה צ"ל 1 , לפי הגדרת המכונה
- לחילופין, המספר שהתקבל עד כה הוא x והשארית היא 1 , וכעת מגיעה הסיבית הבאה
 - כמו לעיל, המספר החדש שווה בערכו ל- $2x$ (כי x הוזז שמאלה בסיבית אחת) ועוד 0 (אם הסיבית החדשה היא 0) או ועוד 1 (אם הסיבית החדשה היא 1)
 - השארית של חלוקת $2x$ ב-5 היא 2 (מדוע? דוגמה?)
 - השארית של חלוקת $2x+1$ ב-5 היא 3
- וכן הלאה, במקרים של שארית $2,3,4$

מכונה לחישוב השארית



- **אזהרה**: זו מכונת מילי...
- אתגר: נניח ש-lsb נכנס ראשון. האם עדיין ניתן לממש כמכונת מצבים סופית?

Summary

- FSM timing – Setup, hold, cycle time / clock frequency
- Clock skew – extending the cycle
- FSM limitations – it's finite