EE 044252: Digital Systems and Computer Structure

Spring 2018

Lecture 8: Communications

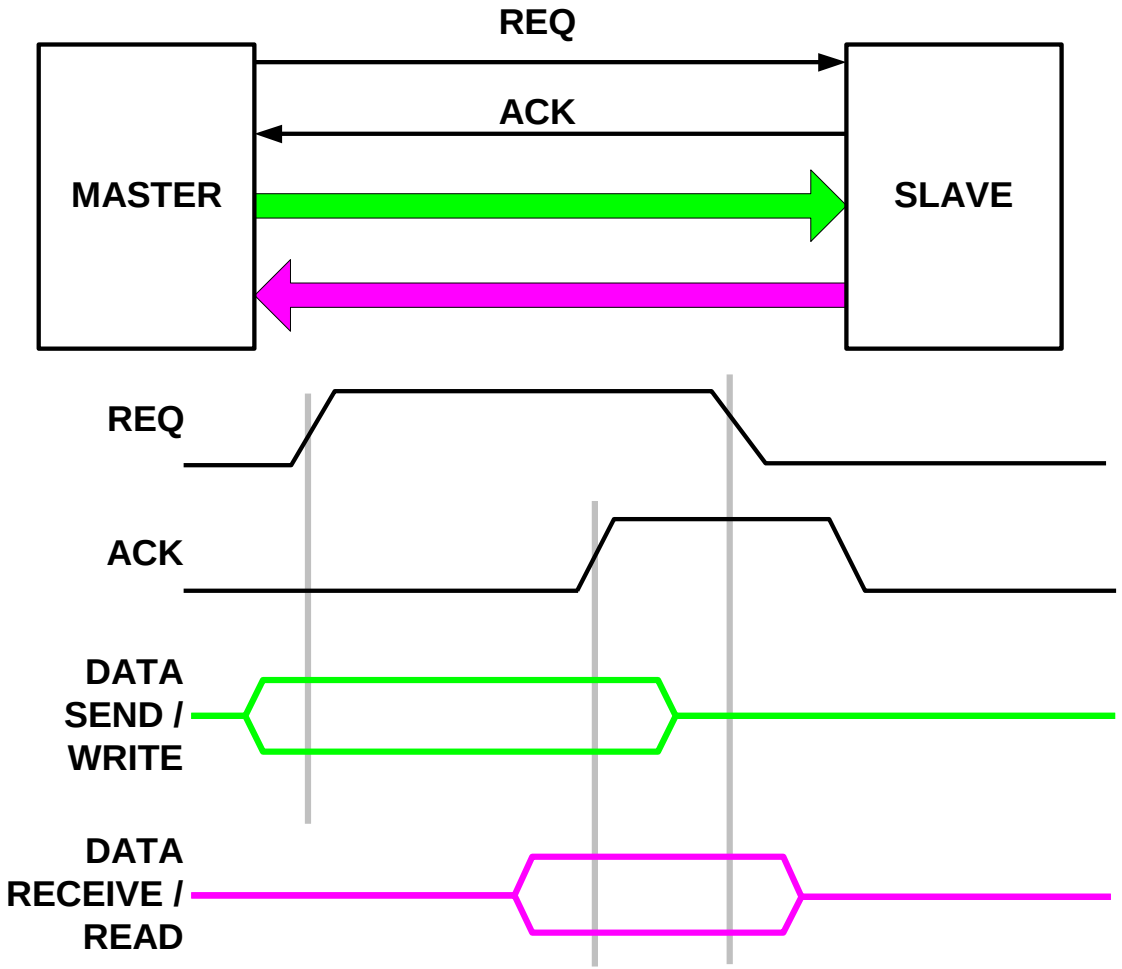
EE 044252: Digital Systems and Computer Structure

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

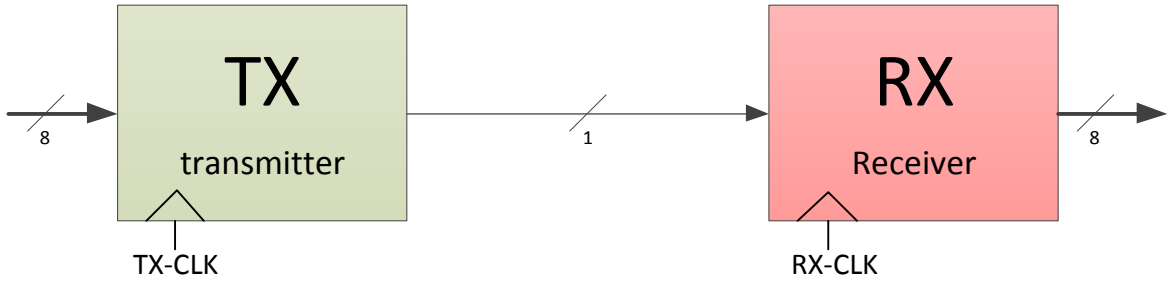
- Serial communication

תקשורת אסינכרונית

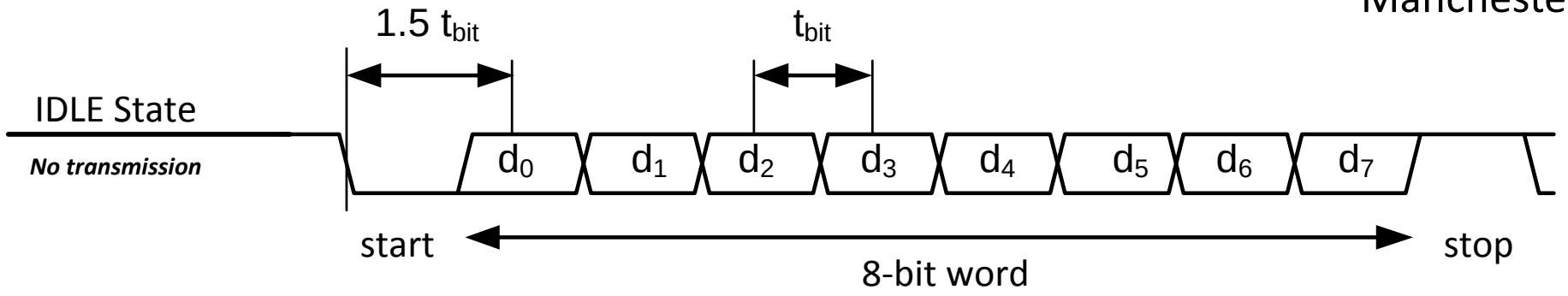


- שני חוטי בקרה ועוד N חוטי נתונים
- פרוטוקול 4-phase
- Event-driven
- אין צורך בשעון. לפעמים לא סביר להשתמש בשעון
- Master – מי שיוזם, מי ששולח Request
- Slave – מי שמחזיר Acknowledgement

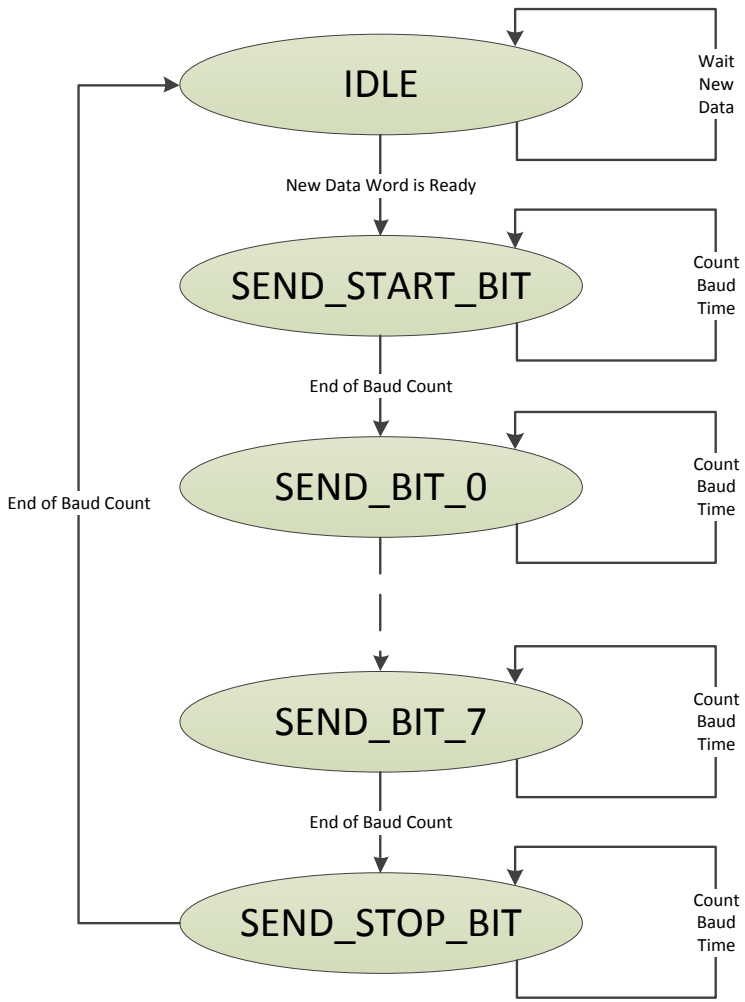
תקשורת על חוט יחיד

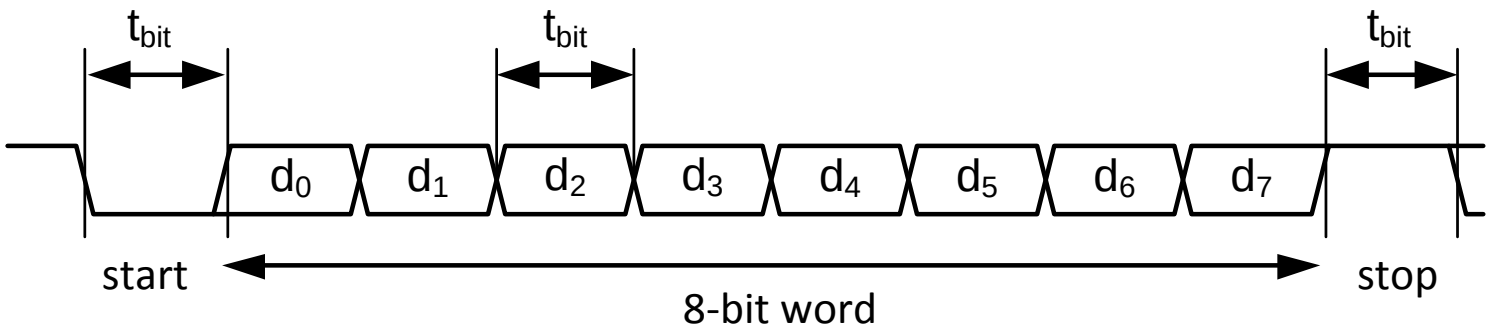


- חוט טלפון, רשת מקומית, ...
- במקום חוטי REQ, ACK:
  - נשתמש בשעון איטי הרבה יותר מהשעונים הפועלים בשני הקצוות
  - נסנכרן אותם מידי פעם
- גם זו קרויה "תקשורת אסינכרונית" אבל בעצם היא קצת סינכרונית
- פרוטוקול תקשורת: **UART**: Universal (Synchronous) Asynchronous Receiver/Transmitter
- נהוג ביציאת COM של ה-PC
- אפשרות אחרת (נהוגה ברשת מקומית מהירה): נקודד את השעון יחד עם הנתונים, למשל "Manchester Code"



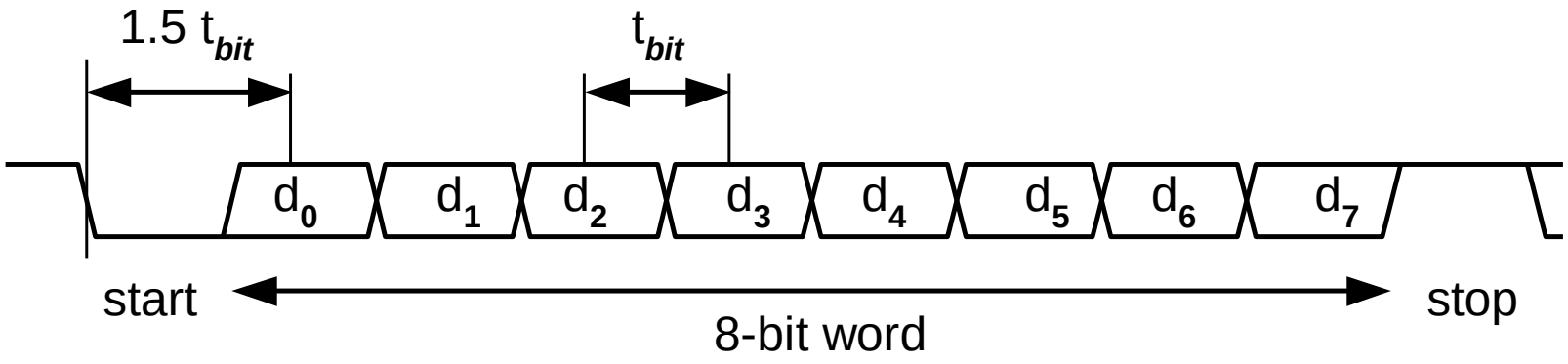
UART – Transmitter FSM





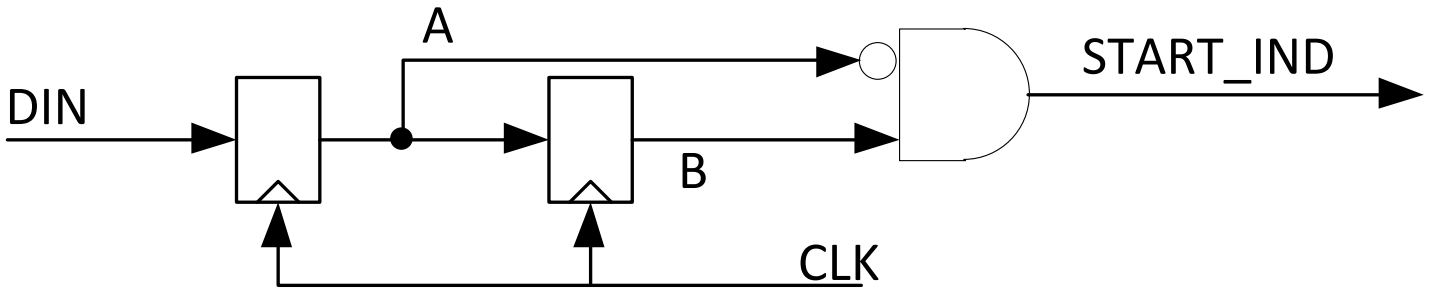
- Baud time
  - Time of single bit transmission (t_{bit})
  - Usually multiple cycles of TX local clock
  - "Baud rate" – UART throughput, e.g. 9,600 bps
- TX FSM can employ a separate counter for Baud Time counting
- Data is saved inside a register and is MUXed out according to FSM state
- Amendments: parity bit, 2-3 stop bits

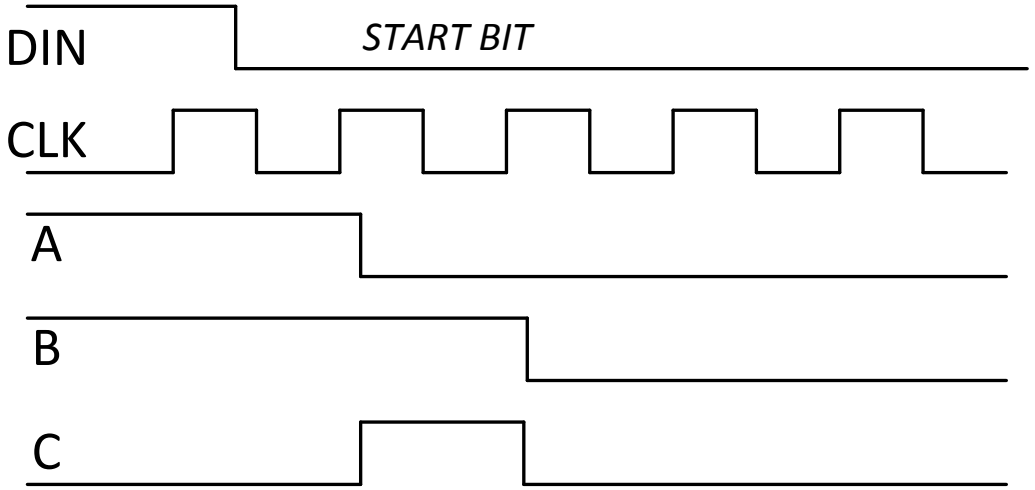
UART - Receiver



- Detect Start of transmission (falling edge detection)
- Count 1.5 x t_{bit} till first data sample
- Count 1 x t_{bit} till each next sample, including STOP bit
- Check out STOP bit value for error detection

UART – Receiver: Start detection



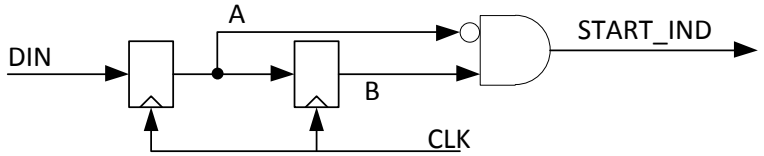


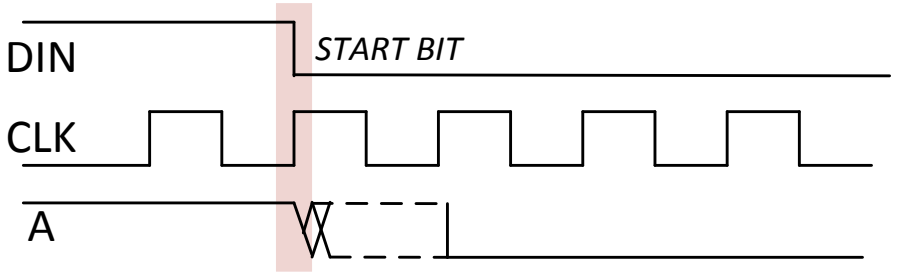
"Edge-detector"

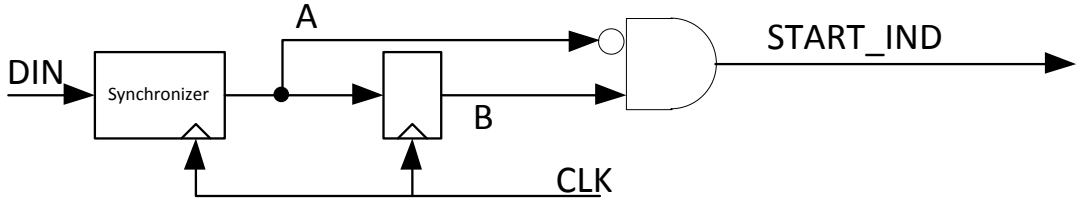
@Home, draw a scheme for:

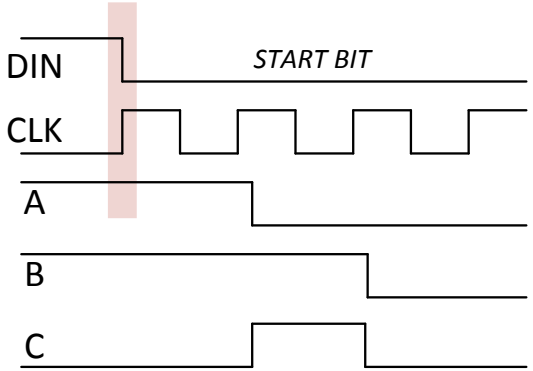
- Edge detector for rising edge
- Edge detector for both edges

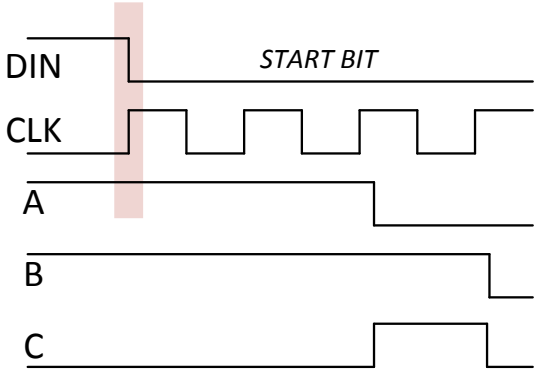
UART – Receiver: input synchronization



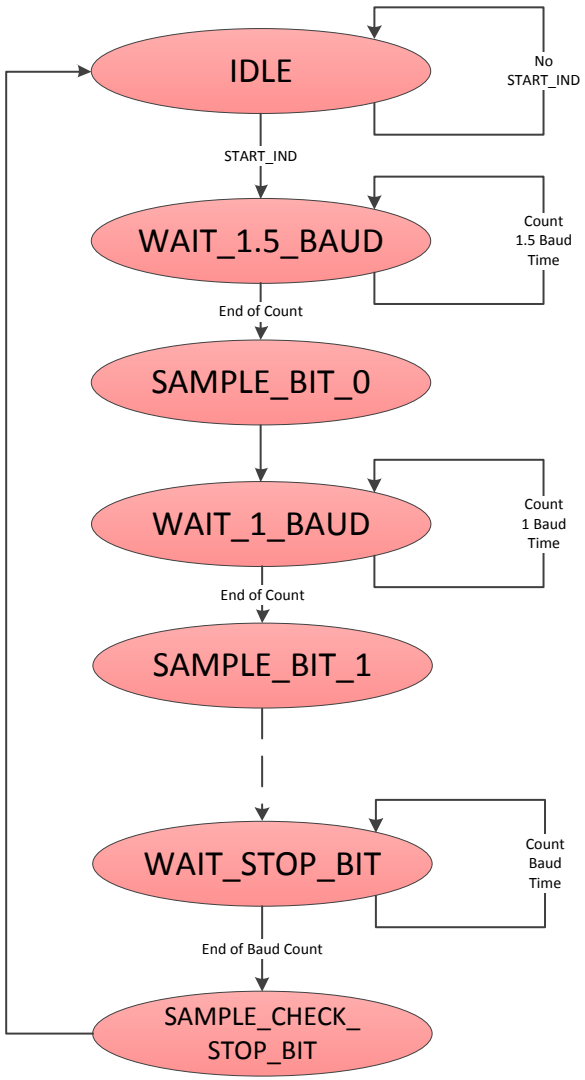


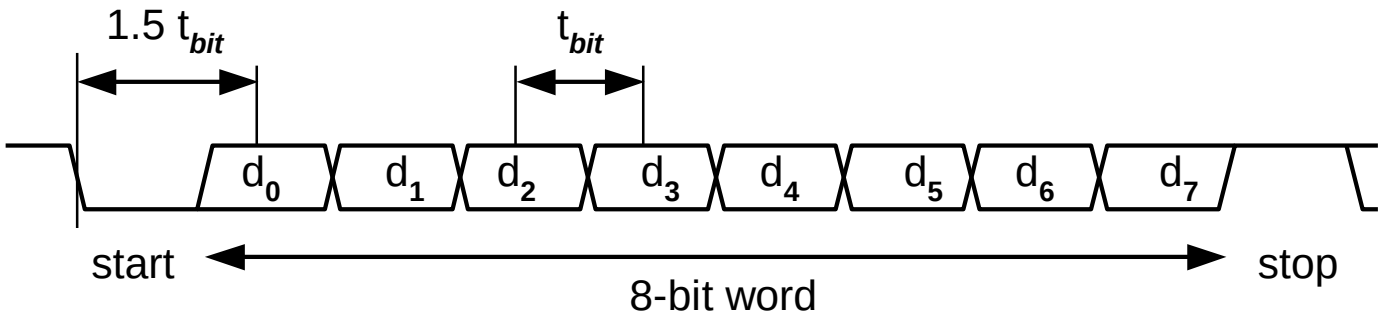




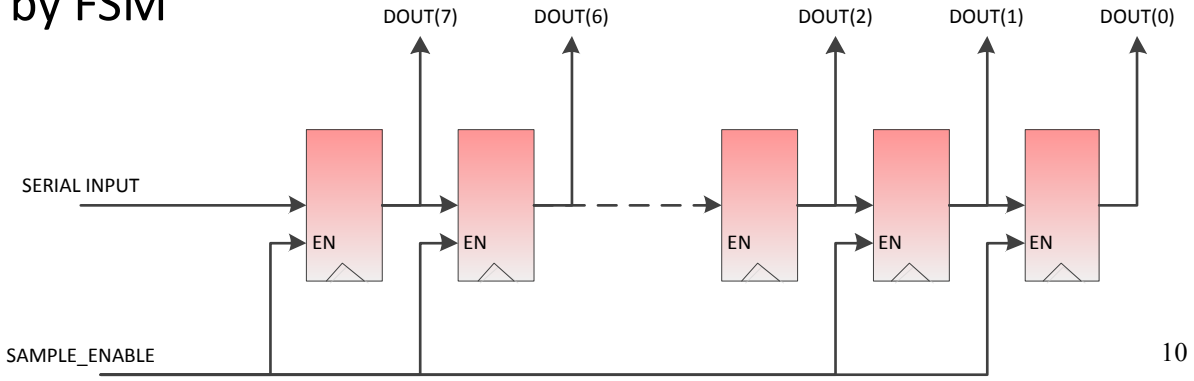


UART – Receiver: RX FSM



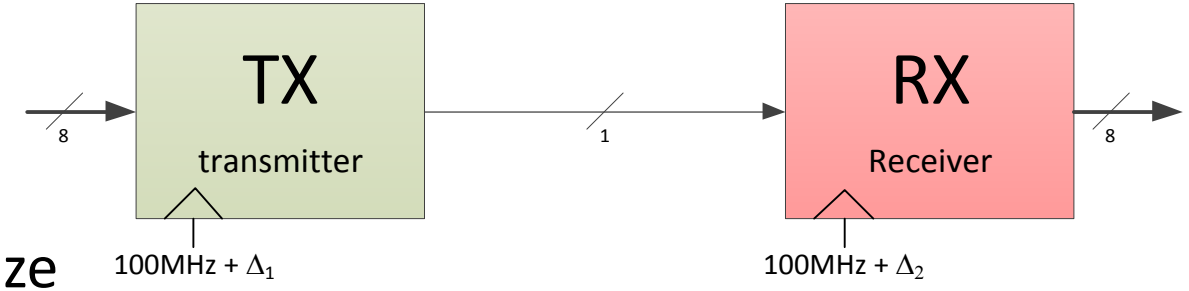


- FSM waits for START bit indication
- RX FSM can employ a separate counter for Baud Time count
- Data is shifted into a SHIFT-REGISTER
  - Enabled by FSM

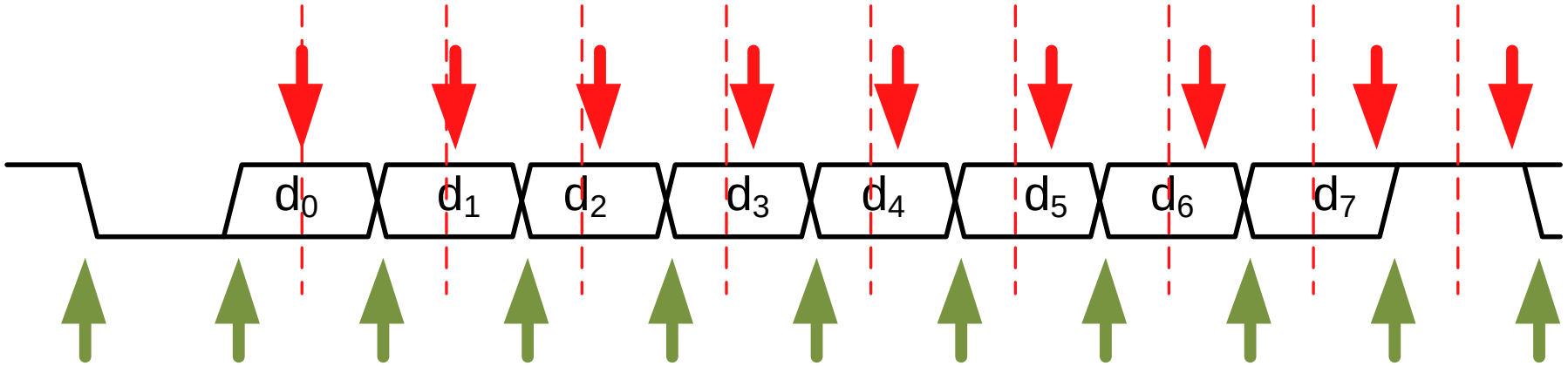


UART – Clock Drift Effect

- TX and RX clocks are not exactly the same
- Measure t_{bit} differently
- Limiting the maximal "character" (word) size
- The drift is accumulated from zero per each START bit

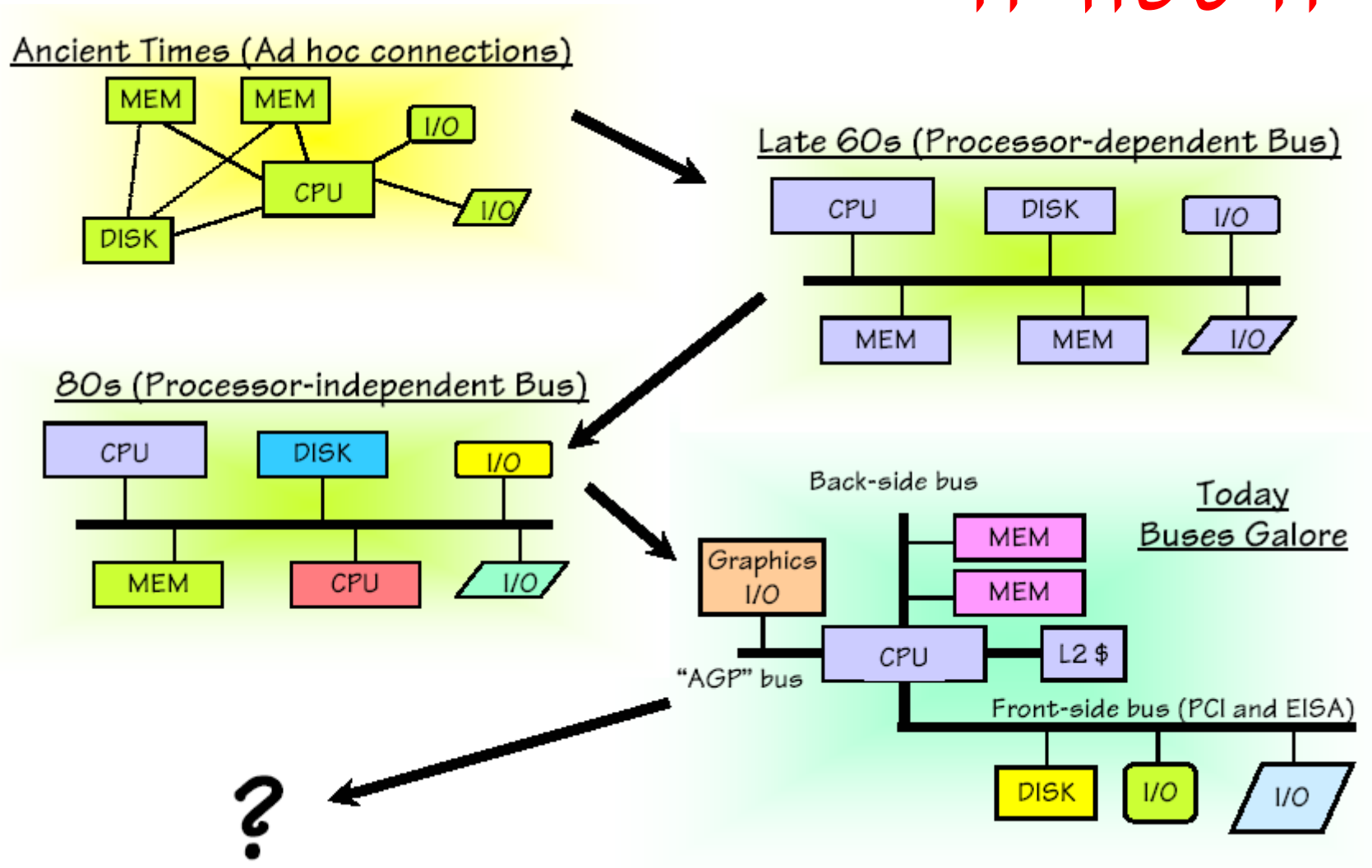


RX Baud Time Count (sampling)



TX Baud Time Count (sending)

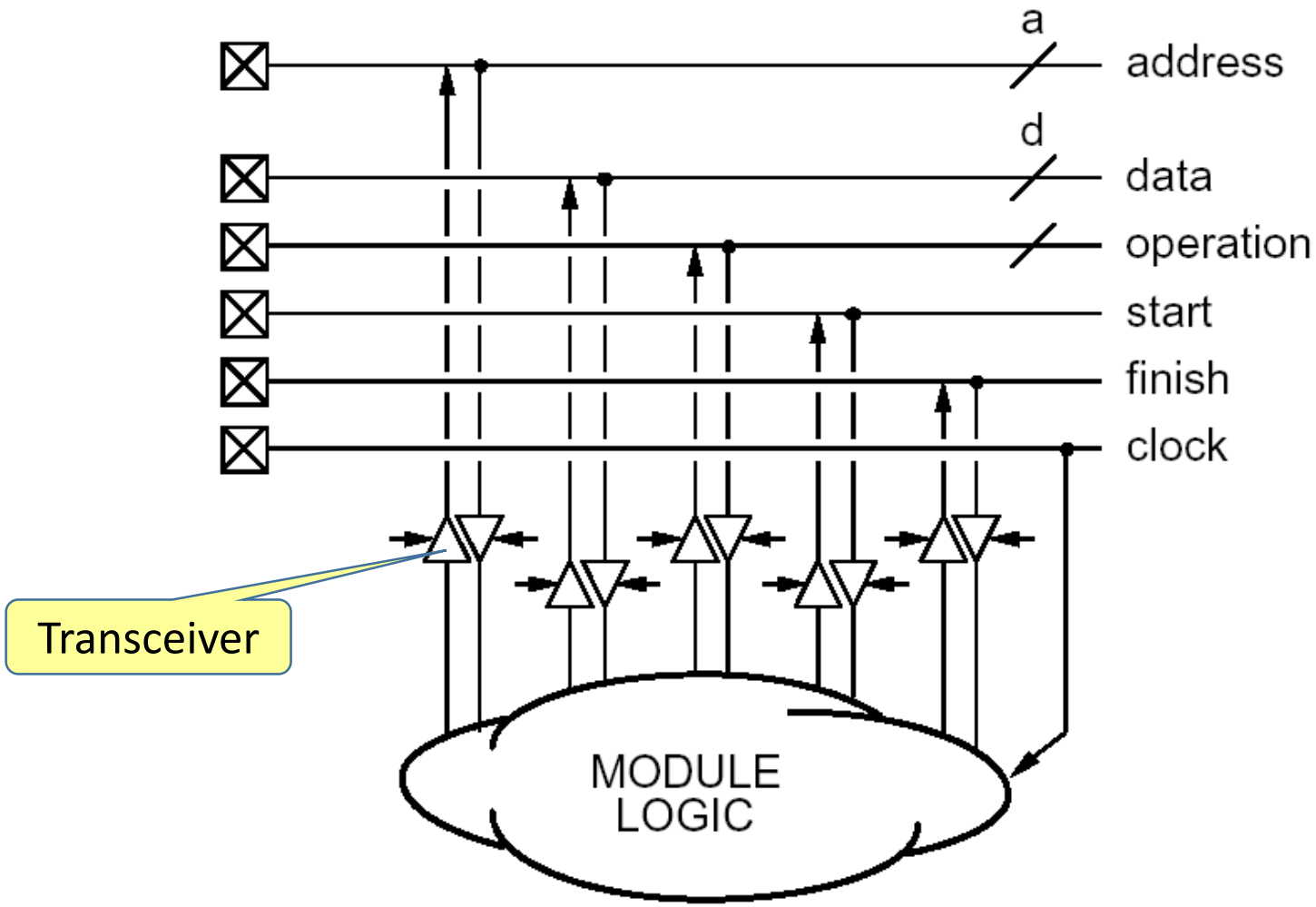
היסטוריה – Buses



Processor-Specific Buses

- עד שנות ה-80
- ISA – ה-BUS הראשון של המחשב האישי
  - המעבד: אינטל 8088
  - ה-BUS: תואם לפינים של המעבד
    - מימוש באמצעות חוצצים להגברת האותות
- לפני המחשב האישי: כל חלקי המחשב נבנו על ידי אותו יצרן (או "תואמים")
- מהפכת המחשב הפתוח:
  - אביזרים נבנו על ידי ספקים רבים ושונים
  - הלקוחות לא רוצים להחליף את כל האביזרים כל פעם שמחליפים מחשב

מה כולל Bus ?


a
address
d
data
operation
start
finish
clock
Transceiver
MODULE
LOGIC

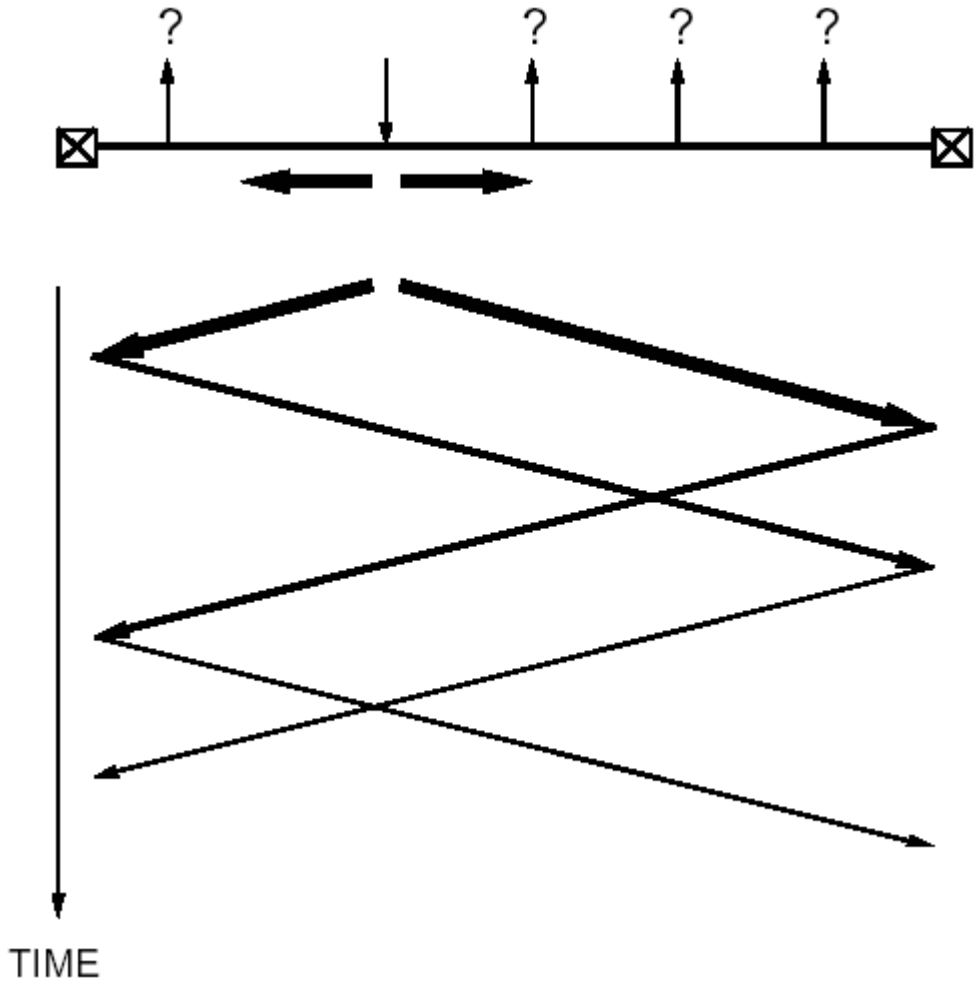
הגדרות: Processor-Independent Buses

- **פעולה (Transaction)** – העברת נתונים על ה-Bus כולל כל פעולות הבקרה הנדרשות
- **Bus Master** – מי שיוזם את הפעולה
  - יתכנו מספר masters אבל רק אחד מהם יכול לפעול כ-master על ה-bus ברגע נתון
- **Bus Slave** – מי שמגיב ל-master
- **Bus Cycle** – זמן הפעולה: מתחילת הבקשה ע"י ה-master ועד לסיום מילוי הבקשה
  - יתכן וידרשו מספר מחזורי שעון

מה כל כך מיוחד ב-Bus ?

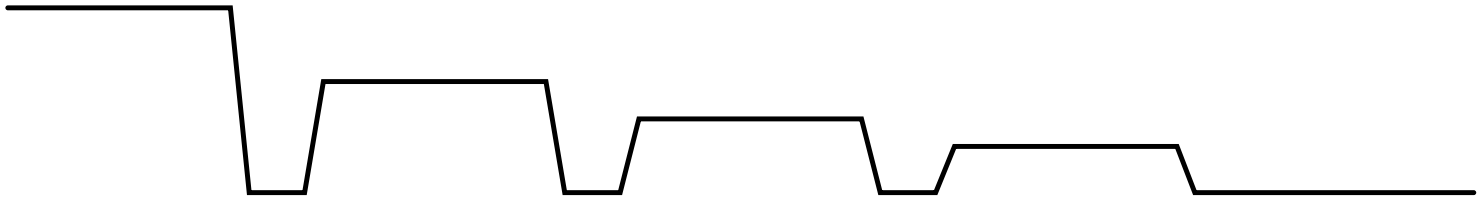
- האם ה-Bus איננו אלא מעגל ספרתי עם חוטים ארוכים?
- "תיאוריה" של חוטים במעגלים ספרתיים (עד כה):
  - המתח (והערך הלוגי) על החוט שווה לכל אורכו
  - הערך הלוגי מתקדם בחוט בזמן אפסי
  - כל ההשהייה במעגל נגרמת ברכיבים ולא בחוטים
  - מעגל כזה קרוי "מקובץ" (lumped)
- תיקון במעגלים מבוזרים
  - מהירות ההתפשטות בחוט סופית
  - אורך החוט גדול: זמן ההתפשטות ארוך יחסית לזמני המיתוג
  - בו-זמנית קורים דברים שונים במקומות שונים לאורך החוט

חוטי ה-Bus הם "קווי תמסורת"

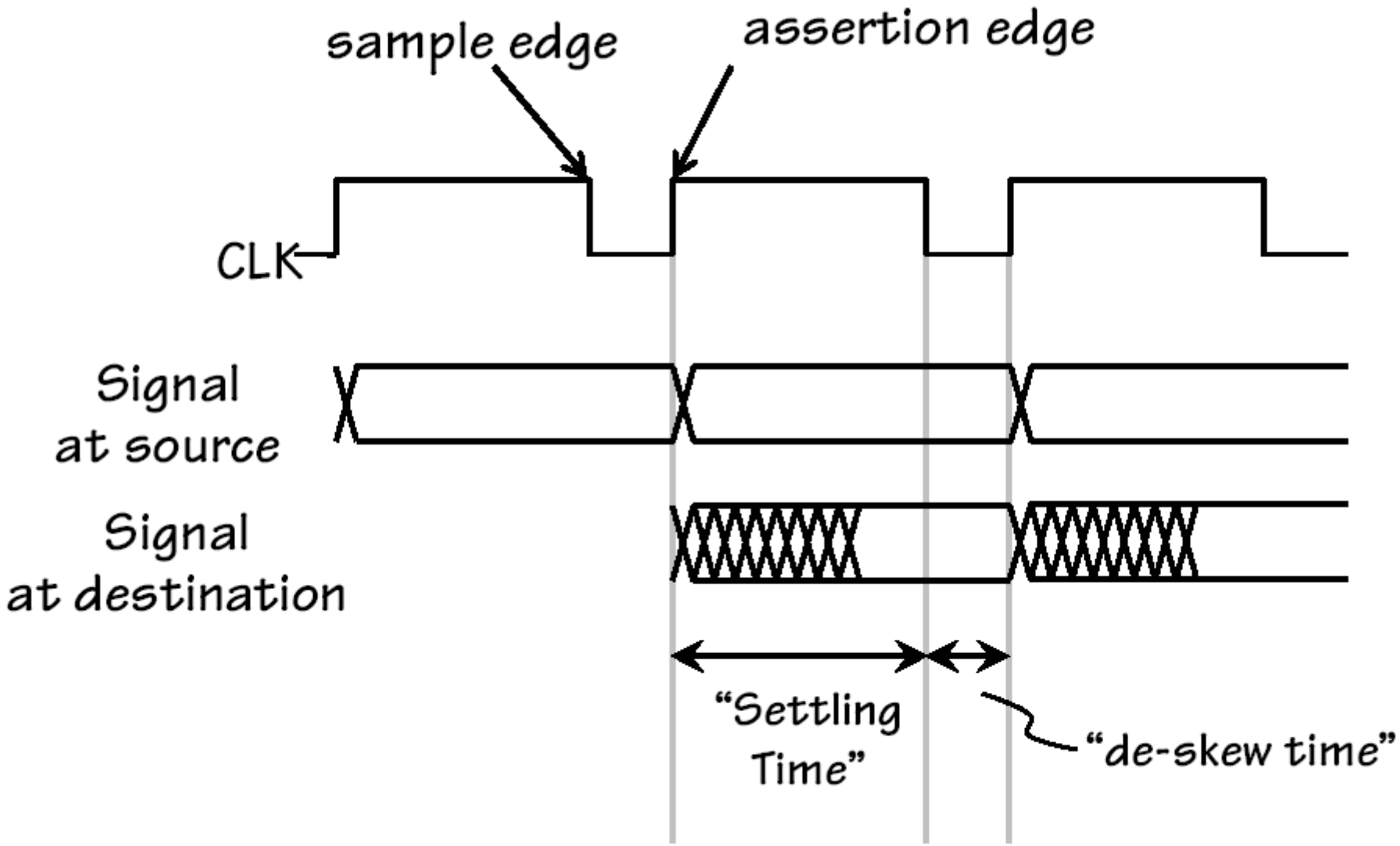


- מהירות האור:
  - 30 ס"מ בננו-שנייה
  - 15 ס"מ\נ"ש בחוט
- Skew לאורך הקו
- החזרות מקצוות לא אידיאלים
  - ההדים מתחברים זה לזה
  - נוצר "צלצול" ונראים glitches בכניסות רכיבים
  - יש להמתין לדעיכת הצלצול לפני שדוגמים את הקו

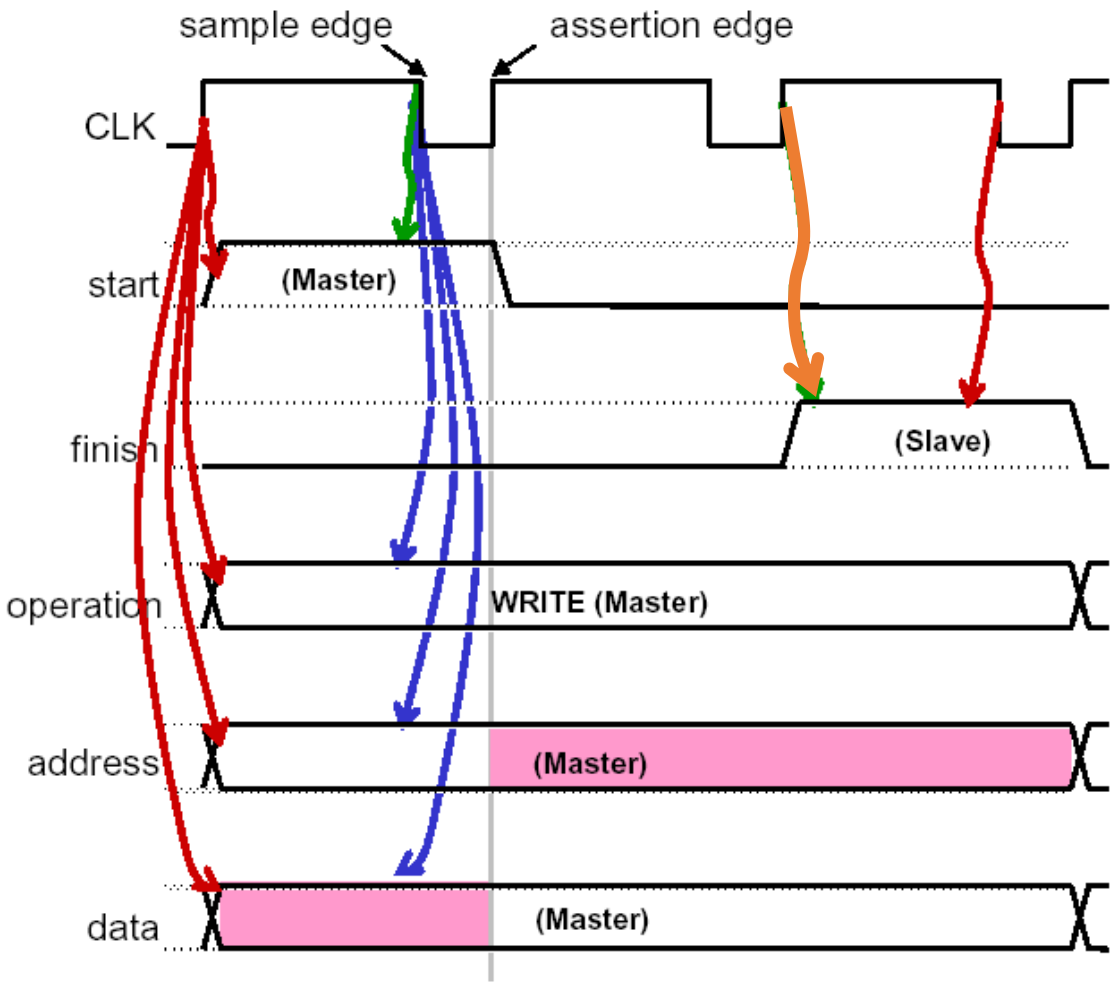
"צלצולים"



- צריך להמתין עד לסיום הצלצול
- לא מתאים לתקשורת אסינכרונית מבוססת REQ, ACK
- פתרון:
  - שעון Bus איטי מספיק
  - שני מאורעות:
    - כתיבת ערך חדש ל-Bus
    - קריאתו, לאחר המתנה לדעיכת הצלצולים

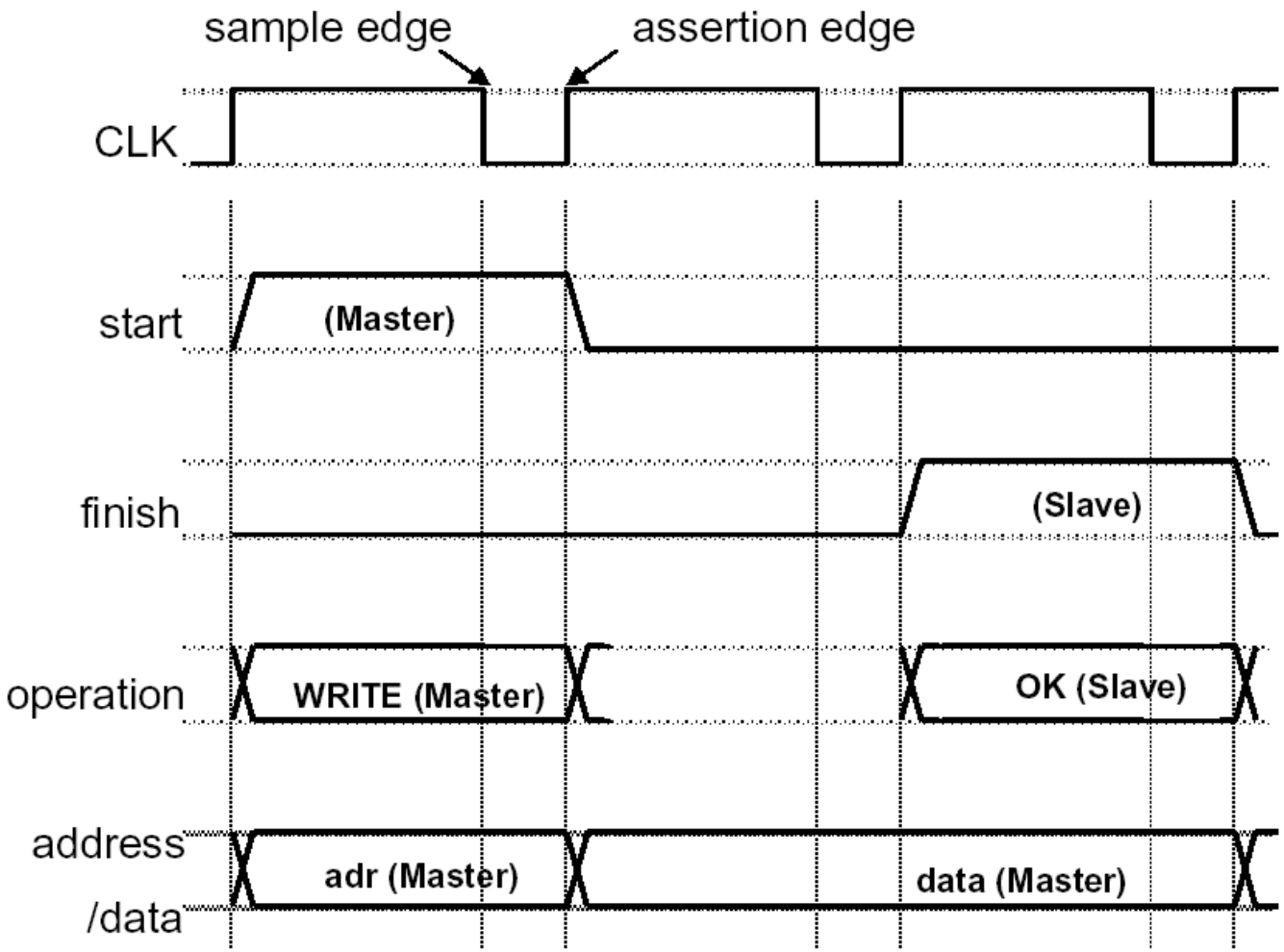

sample edge
assertion edge
CLK
Signal
at source
Signal
at destination
"Settling
Time"
"de-skew time"

פעולת Bus פשוטה



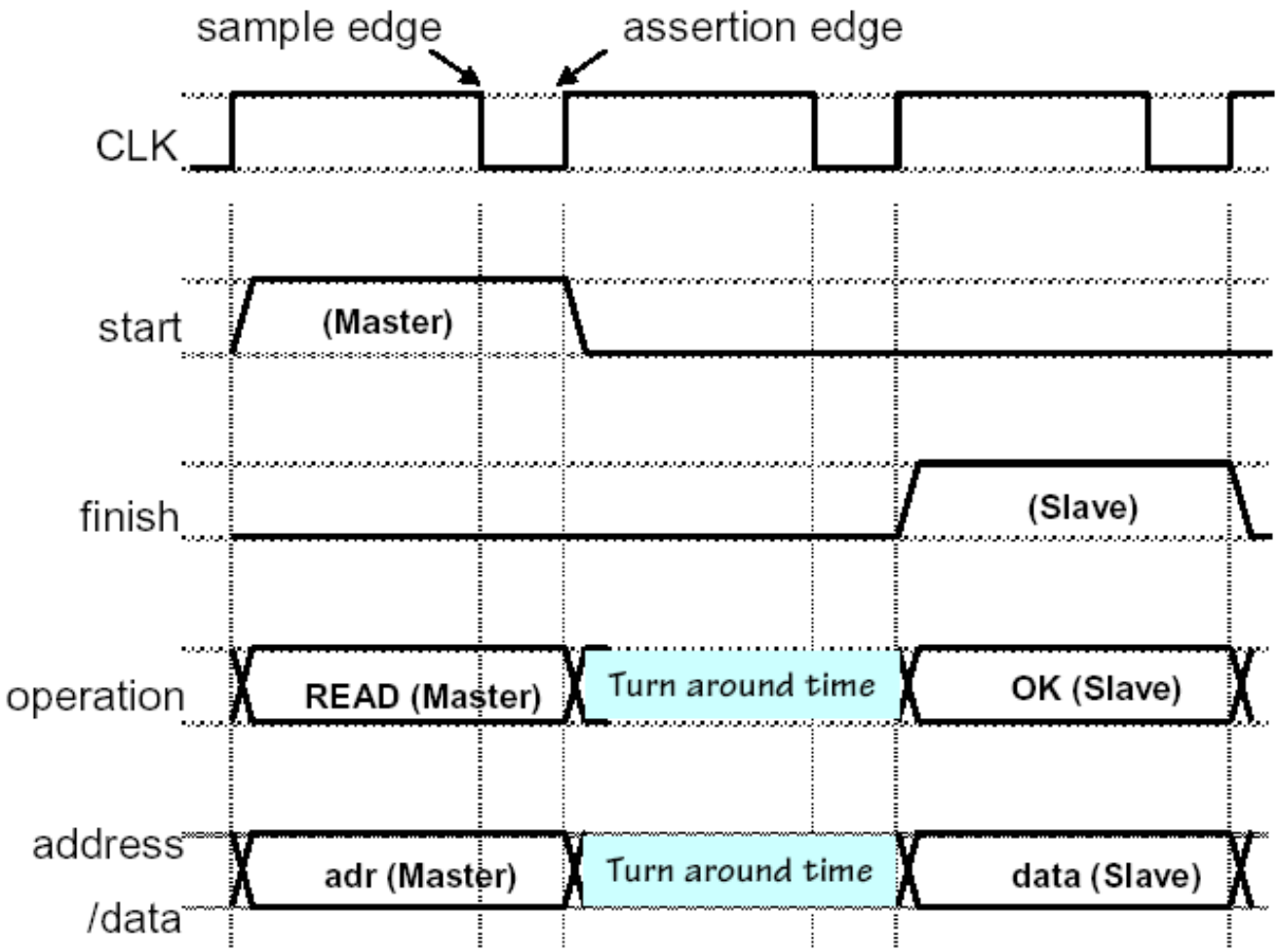
- Master
  - קובע מה תהיה הפעולה
  - כותב כתובת ו-start
  - ממתין ל-slave
- Slave
  - ממתין ל-start
  - בודק כתובת
  - אם עבורו אז
    - בודק מה הפעולה
    - מבצע אותה
    - מסמן finish
  - יכול לבקש יותר זמן (wait cycles)
- Bus
  - ממתין ל-start
  - מתחיל מונה
  - אם אין תגובה לאחר המניה, צועק

Multiplexed Bus: Write Transaction



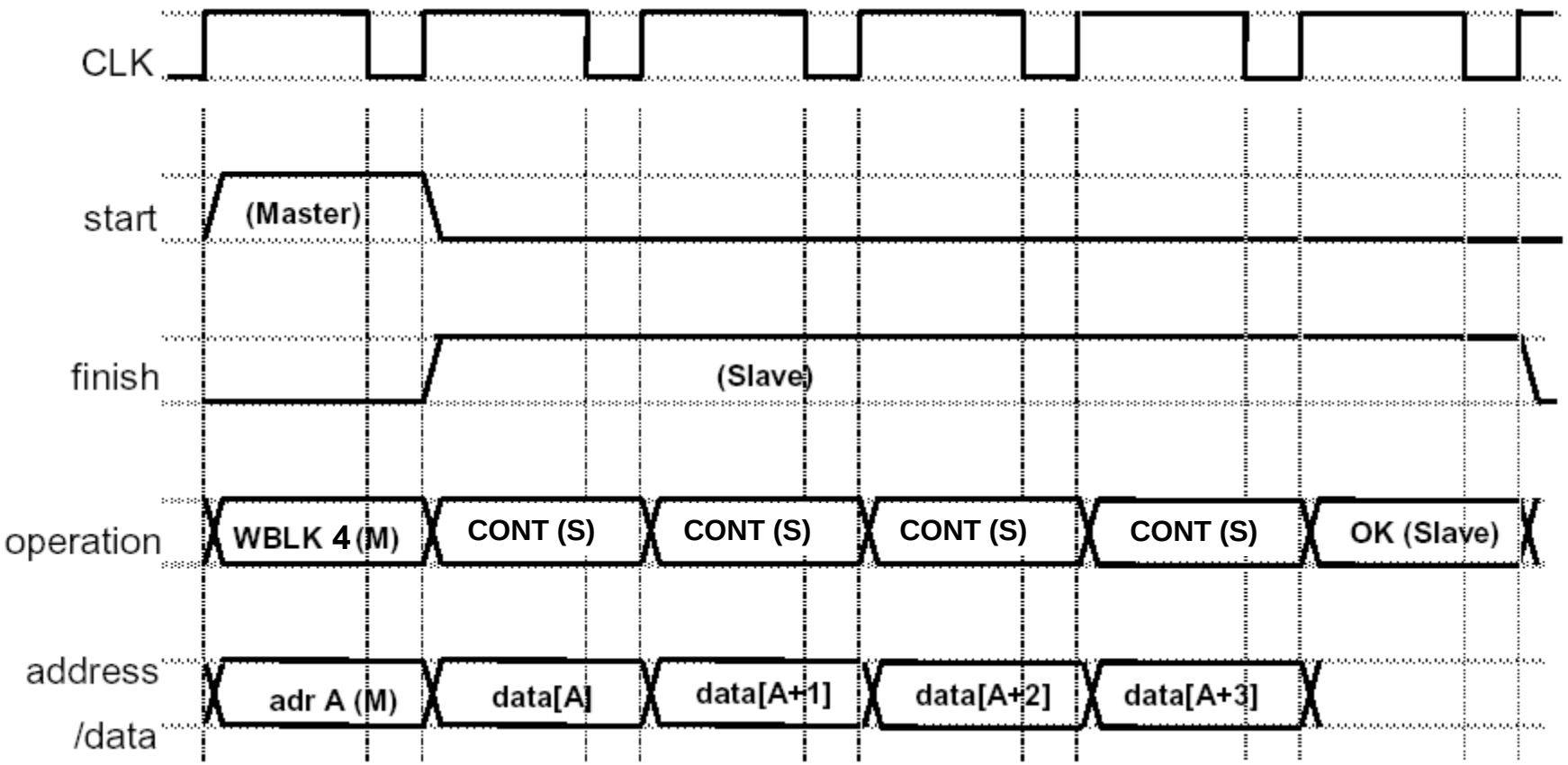
- חסכון בחוטים
- מתאים ל-slave שקודם צריך כתובת ורק אחר כך נתונים

Multiplexed Bus: Read Transaction



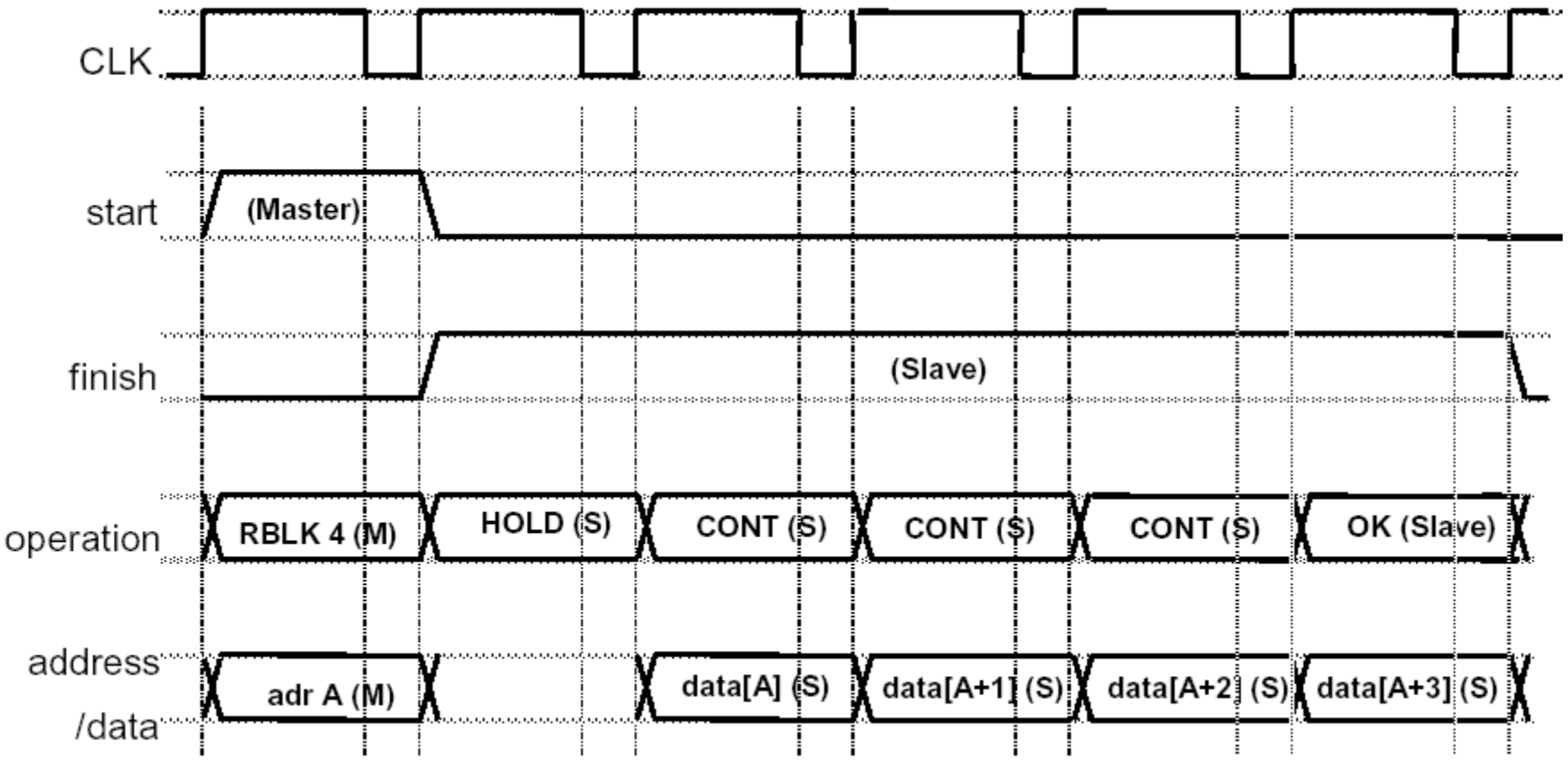
Throughput: 3 Clocks/word

Block Write Transfers

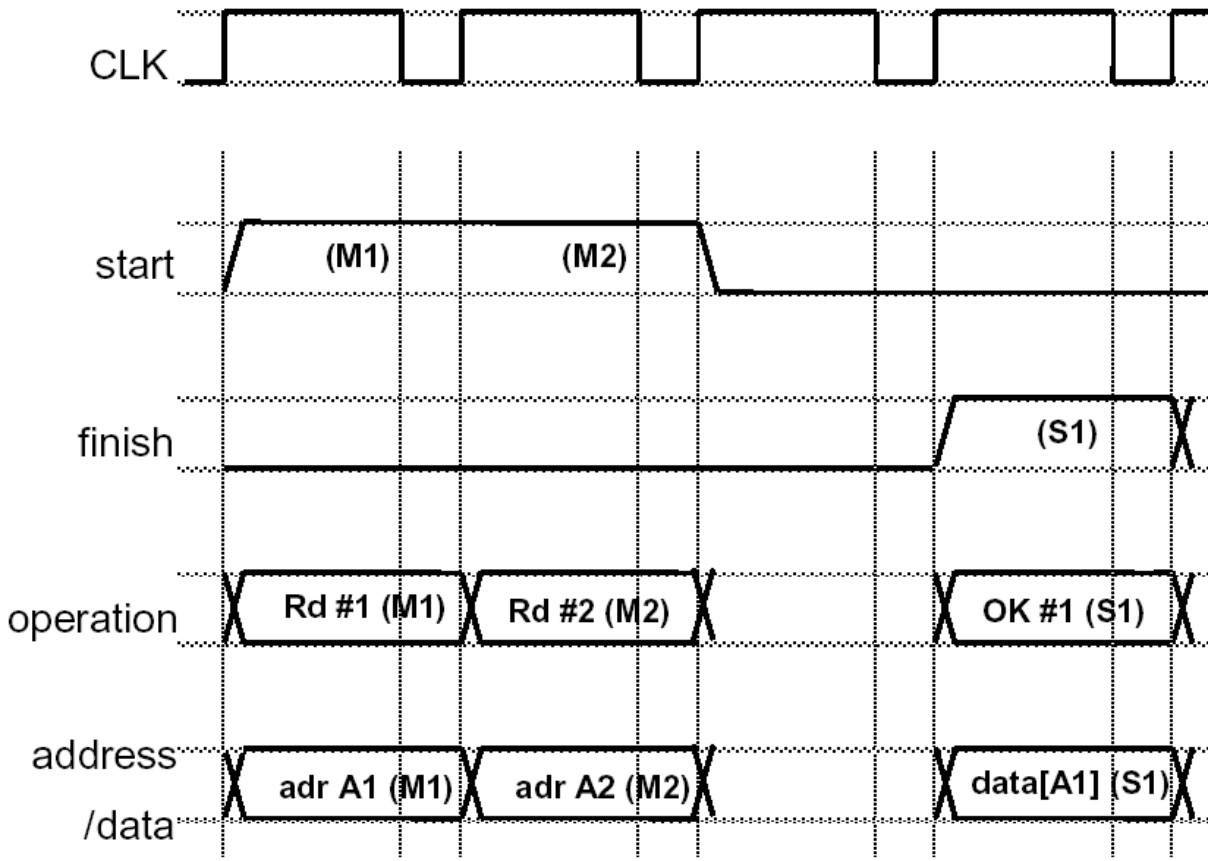


- יעיל להעברת בלוקים—throughput מתקרב למילה כל מחזור
- ה-slave חייב לייצר כתובות עוקבות

Block Read Transfers



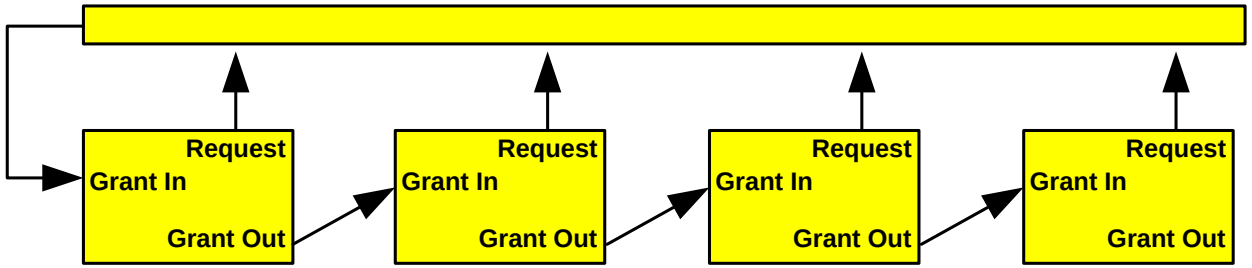
Split-Transactions



- מזכיר pipelines
- Master שולח כתובת, קוד פעולה
  - ואולי ממשיך לשלוח כתובות נוספות
- רק מאוחר יותר מתקבלת התוצאה מה-slave
- Throughput מילה לשני מחזורים, לא תלוי ב-latency
- פרטים, פרטים..

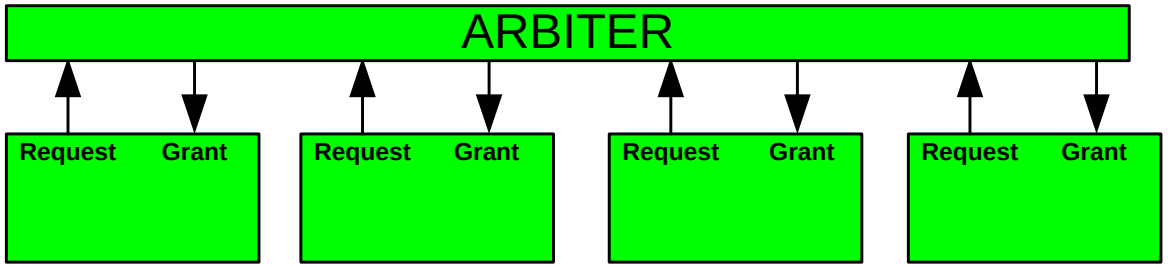
Bus Arbitration: Multiple Bus Masters

Daisy Chain:



- Daisy Chain: פשוט, אך לא גמיש. עדיפות לפי המקום בשרשרת
- Arbiter מקבילי: יקר יותר (לוגריתמי) אך גמיש / ניתן לתכנות
- שיקולים: הגינות מול עדיפות, משך המתנה מקסימלי, גמישות בתכנון ותכנות, פשטות / מחיר

Parallel Arbitration:



Summary

- Serial communication: Async, UART
- Buses