

Digital Logic Systems

Recitation 9: Shifters, Binary Addition

Guy Even Moti Medina

School of Electrical Engineering Tel-Aviv Univ.

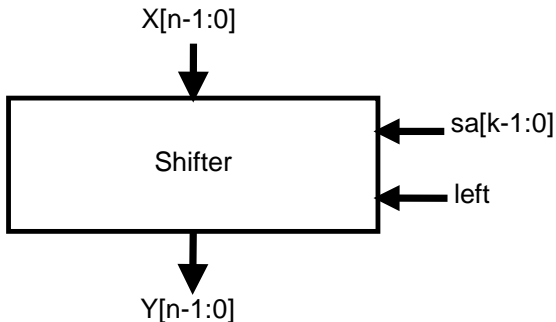
May 14, 2019

- ① General Note: YOU ARE WELCOME AT THE “TIGBUR” HOUR EVERY TUESDAY 17:00 AT CLASS 134 WOLFSON BUILDING.

Shifters - ports

Generally, all shifters have the following ports:

- ① $x[n-1:0]$ - input string
- ② $y[n-1:0]$ - output string
- ③ $sa[\log n - 1:0]$ - “shift-amount”, controls the amount of bits to be shifted
- ④ $left$ - input determines the direction of the shift to be left. (for bi-directional shifters)



We saw 3 types of shifts and different circuits that implemented them.

- 1 **Cyclic** shifts - "bits pushed from one side of the string, appear on the opposite edge". $\text{BARREL-SHIFTER}(n)$ consists of a series of CLS
- 2 **Logical** shifts - "bits that are pushed from the string boundaries - are lost". We saw L-SHIFT which can be implemented by cascading LLS or LRS circuits.
- 3 **Arithmetical** shifts - "same as logical, except for shifting right, the MSB is duplicated". The implementations of ARS and ARITH-SHIFT are left for your home assignment.

Shifters - Tips

- ① Shifters can be used for fast **division/multiplication**, as well as for **masking**.
- ② As always, we assume **free wiring** in our course
- ③ **Bit reversal** enables us to apply left shifts to obtain right shift and vice versa.
- ④ **Modular Design of a Shifter** - We break the sa into its powers of two. Specific circuits (CLS, LLS, ARS) implement these power-of-2 shifts. Finally we **cascade** these modules to obtain the overall shift.
- ⑤ You might have already noticed, that signals (inputs/outputs) can be classified into **data-signals** and **control-signals**.

L-SHIFT(8) example in Logisim

Pay attention to

- ① Modular design - top-down approach
- ② Hardwiring - wires help us implement shifts, for free!
- ③ Usage of new wiring tools in Logisim: *tunnels* and *splitters*.

L-SHIFT(8) example in Logisim

Definition

A $\text{L-SHIFT}(n)$ is a combinational circuit defined as follows:

Input:

- $x[n-1:0] \in \{0,1\}^n$,
- $sa[k-1:0] \in \{0,1\}^k$, where $k = \lceil \log_2 n \rceil$, and
- $\ell \in \{0,1\}$.

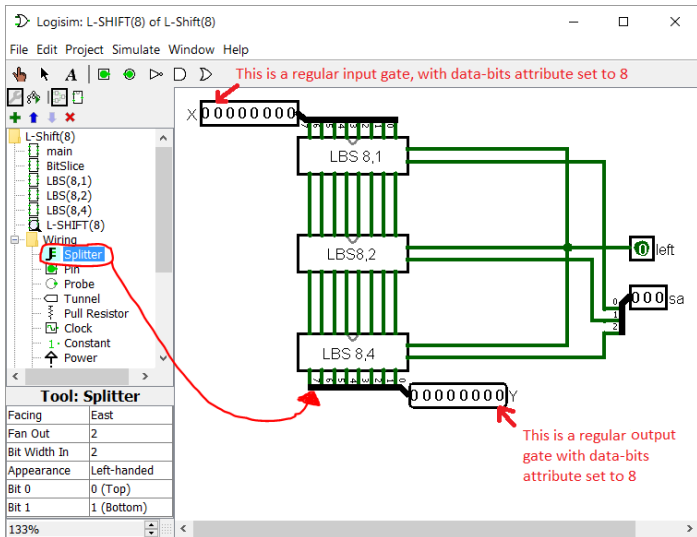
Output: $y[n-1:0] \in \{0,1\}^n$.

Functionality: The output \vec{y} satisfies

$$\vec{y} \triangleq \begin{cases} \text{LLS}(\vec{x}, \langle \vec{sa} \rangle) & \text{if } \ell = 1, \\ \text{LRS}(\vec{x}, \langle \vec{sa} \rangle) & \text{if } \ell = 0. \end{cases}$$

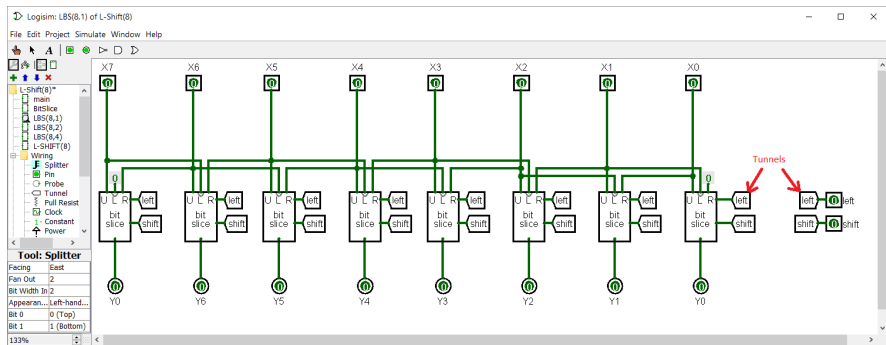
L-SHIFT(8) example in Logisim - Top-Down approach

Let us design the top level, consisting of 3 sub-circuits
 $LBS(8,1), LBS(8,2), LBS(8,4)$.



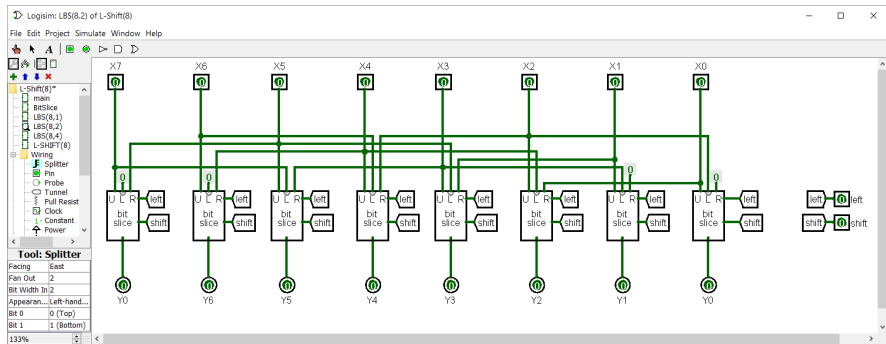
L-SHIFT(8) example in Logisim - sub module LBS(8,1)

Wise wiring - implements the shift.



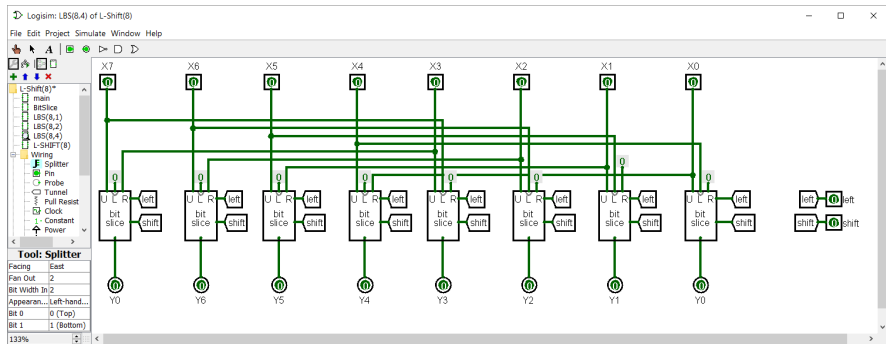
L-SHIFT(8) example in Logisim - sub module LBS(8,2)

Again, smart hardwiring is the key



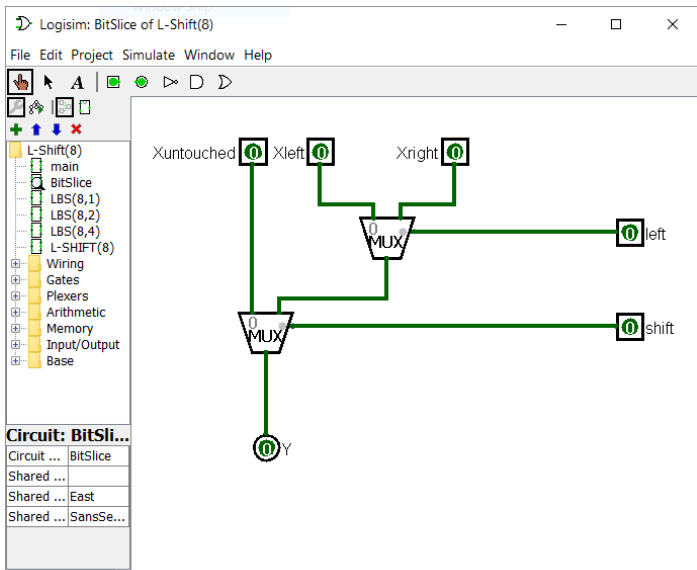
L-SHIFT(8) example in Logisim - sub module LBS(8,4)

And again, smart hardwiring is the key



L-SHIFT(8) example in Logisim - “bit slice”

This is the basic building block, the lowest design level in our project.



Binary addition warm-up:

- ① We know how to represent **natural numbers**
- ② We would like to know how to **add** them together
- ③ Adding two binary-represented numbers is **easy**.
- ④ We saw **3 different implementations** of binary adder

Binary-Adder Logic Basics

Definition

$\text{ADDER}(n)$ - a **binary adder** with input length n is a combinational circuit specified as follows.

Input: $A[n-1:0], B[n-1:0] \in \{0,1\}^n$, and $C[0] \in \{0,1\}$.

Output: $S[n-1:0] \in \{0,1\}^n$ and $C[n] \in \{0,1\}$.

Functionality:

$$\langle \vec{S} \rangle + 2^n \cdot C[n] = \langle \vec{A} \rangle + \langle \vec{B} \rangle + C[0]. \quad (1)$$

Definition (Full-Adder)

FA - a **Full-Adder** is a combinational circuit with 3 inputs $x, y, z \in \{0,1\}$ and 2 outputs $c, s \in \{0,1\}$ that satisfies:

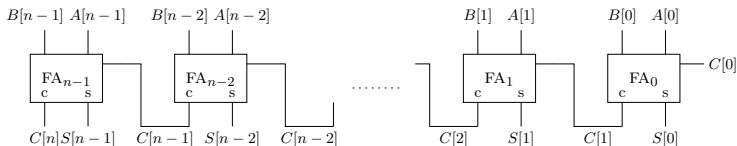
$$2c + s = x + y + z.$$

Claim

$$s = x \oplus y \oplus z$$

$$c = (x \cdot y) \vee (y \cdot z) \vee (x \cdot z)$$

Implementation 1 - Ripple Carry Adder RCA(n)



- Same addition algorithm used for adding numbers by hand.
- Row of n Full-Adders connected in a chain.

Claim

For each $0 \leq i \leq n-1$, the cone of Boolean functions corresponding to $C[i+1]$ and $S[i]$ consists of $2i+3$ inputs corresponding to $A[i:0], B[i:0]$, and $C[0]$.

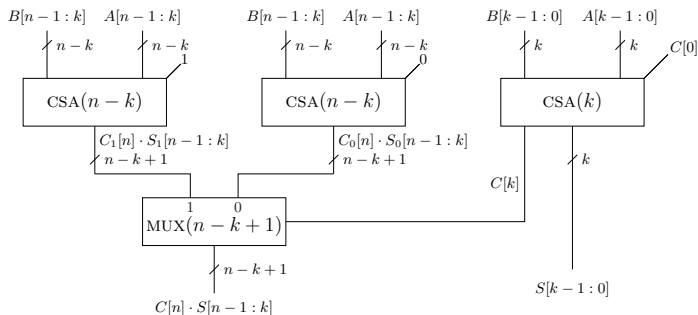
Performance

- $c(\text{RCA}(n)) = n \cdot c(\text{FA}) = \Theta(n)$.
- $d(\text{RCA}(n)) = n \cdot d(\text{FA}) = \Theta(n)$.

Implementation 2 - Conditional Sum Adder $\text{CSA}(n)$

basis: A $\text{CSA}(1)$ is simply a Full-Adder.

reduction step:



Performance

- $c(\text{CSA}(n)) = \Theta(n^{\log_2 3})$.
- $d(\text{CSA}(n)) = \Theta(\log n)$.

Implementation 3 - Compound Adder COMP-ADDER(n)

Definition

COMP-ADDER(n) - a **Compound Adder** with input length n is a combinational circuit specified as follows.

Input: $A[n-1:0], B[n-1:0] \in \{0,1\}^n$.

Output: $S[n:0], T[n:0] \in \{0,1\}^{n+1}$.

Functionality:

$$\langle \vec{S} \rangle = \langle \vec{A} \rangle + \langle \vec{B} \rangle$$

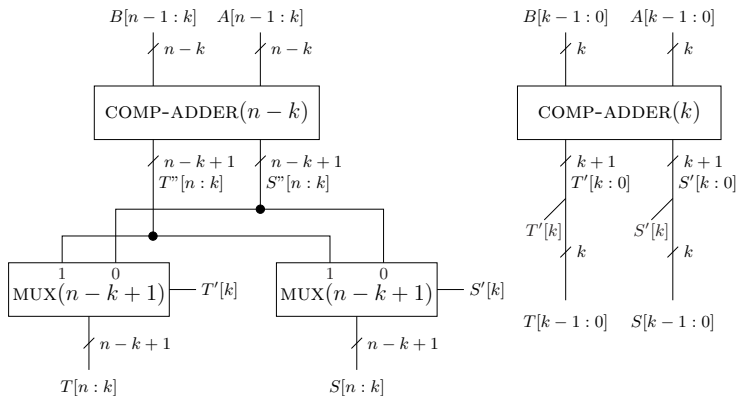
$$\langle \vec{T} \rangle = \langle \vec{A} \rangle + \langle \vec{B} \rangle + 1.$$

Note that a Compound Adder does not have carry-in input. To simplify notation, the carry-out bits are denoted by $S[n]$ for the sum and by $T[n]$ for the incremented sum.

Implementation 3 - Compound Adder COMP-ADDER(n)

basis: $n = 1$, we simply use a Full-Adder and a Half-Adder.

reduction step:



Performance

- $c(\text{COMP-ADDER}(n)) = \Theta(n \log n)$.
- $d(\text{COMP-ADDER}(n)) = \Theta(\log n)$.

Problem on Adders and Shifters

שאלה 1 – מעגל צירופי (35 נק')

להלן המפרט של המעגל הצירופי $C(n)$

קלט: $D[k-1:0] \in \{0,1\}^k$ $Y[n-1:0] \in \{0,1\}^n$ כאשר $n = 2^k$ $X[n-1:0] \in \{0,1\}^n$

פלט: $Z[2n-1:0] \in \{0,1\}^{2n}$

פונקציונליות:

נסמן ב- d את המספר המיוצג על ידי הקלט D בשיטה הבינארית, דהיינו:

$$d \triangleq \langle D[k-1:0] \rangle$$

נסמן ב- z את הביטוי הבא:

$$z \triangleq \langle X[n-1:0] \rangle + \sum_{i=0}^{n-1} Y[i] \cdot 2^{i+n-d}$$

הפלט $Z[2n-1:0]$ חייב לקיים:

$$\langle Z[2n-1:0] \rangle = z$$

Problem on Adders and Shifters

א. (7 נק') בצעו סימולציה לוגית של המפרט של מעגל $C(4)$ על הקלטים:

$$X[3:0] = 1010$$

$$Y[3:0] = 1111$$

$$D[1:0] = 10$$

מלאו את הטבלה בהתאם:

	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
X	אין	אין	אין	אין	1	0	1	0
Y	אין	אין	אין	אין	1	1	1	1
D	אין	אין	אין	אין	אין	אין	1	0
Z								

Problem on Adders and Shifters

א. (7 נק') בצעו סימולציה לוגית של המפרט של מעגל $C(4)$ על הקלטים:

$$X[3:0] = 1010$$

$$Y[3:0] = 1111$$

$$D[1:0] = 10$$

מלאו את הטבלה בהתאם:

	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
X	אין	אין	אין	אין	1	0	1	0
Y	אין	אין	אין	אין	1	1	1	0
D	אין	אין	אין	אין	אין	אין	1	0
Z	0	1	0	0	0	1	1	0

Problem on Adders and Shifters

ב. (8 נק') עבור קלטים X, Y, D מסוימים, ה-Overflow מוגדר כמאורע שבו z לא ניתן לייצוג על ידי $2n$ סיביות. האם מאורע זה ייתכן במקרה שלנו? אם כן – תנו דוגמה עם קלטים ספציפיים. אם לא, הראו ש- z נמצא בתחום המספרים הניתנים לייצוג בשיטה הבינארית בעזרת $2n$ סיביות.

ה-Overflow ייתכן \ לא ייתכן (הקיפו את הנכון)

הסבר:

Problem on Adders and Shifters

ב. (8 נק') עבור קלטים X, Y, D מסוימים, ה-Overflow מוגדר כמאורע שבו z לא ניתן לייצוג על ידי $2n$ סיביות. האם מאורע זה ייתכן במקרה שלנו? אם כן – תנו דוגמה עם קלטים ספציפיים. אם לא, הראו ש- z נמצא בתחום המספרים הניתנים לייצוג בשיטה הבינארית בעזרת $2n$ סיביות.

ה Overflow-ייתכן \ לא ייתכן (הקיפו את הנכון)

הסבר

כדי להראות שה-Overflow לא אפשרי, ניתן להראות שהמספר z ניתן לייצוג על ידי המחרוזת הבינרית Z בעלת $2n$ הסיביות. בשביל זה ניקח את הקלטים D, Y, X שייתנו את ה- z המקסימלי:

$$z_{max} = \max \left(\langle X[n-1:0] \rangle + 2^{n-d} \sum_{i=0}^{n-1} Y[i] \cdot 2^i \right) = \max(\langle X \rangle + 2^{n-d} \langle Y \rangle)$$

בשביל למקסם את הביטוי, ניקח את הקלטים Y, X הגדולים ביותר: $X = Y = 1^n$, ונבחר את $D = 0^k$

$$= 2^n - 1 + 2^n(2^n - 1) = 2^{2n} - 2^n + 2^n - 1 = 2^{2n} - 1$$

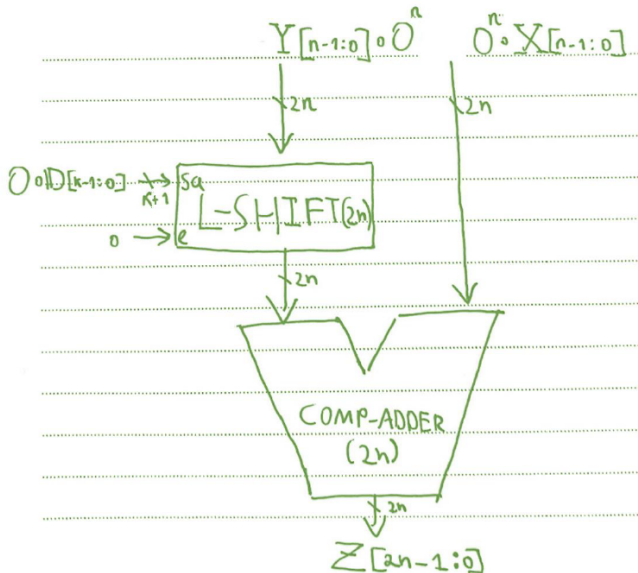
ומכיוון ש- z_{max} זה עדיין ניתן לביטוי על ידי $2n$ הסיביות של המחרוזת Z , אזי ה-Overflow לא ייתכן.

Problem on Adders and Shifters

ג. (15 נק') ממשו את המעגל הצירופי $C(n)$. מטרה: מעגל מהיר ככל האפשר, עם מחיר לכל היותר $O(n \cdot \log n)$.

Problem on Adders and Shifters

ג. (15 נק') ממשו את המעגל הצירופי $C(n)$. מטרה: מעגל מהיר ככל האפשר, עם מחיר לכל היותר $O(n \cdot \log n)$.



Problem on Adders and Shifters

א. (5 נק') בצעו סימולציה לוגית של המעגל $C(4)$ לפי תכנונםם בסעיף ב' עבור הקלט הבא:

(העתיקו את המעגל ורישמו ליד כל ערוץ את ערכו הלוגי)

סימולציית קלט $X[3:0]=1010$ $Y[3:0]=1111$ $D[1:0]=11$

