Digital Logic Systems

Recitation 9: Shifters & Addition

Guy Even Moti Medina

School of Electrical Engineering Tel-Aviv Univ.

January 3, 2012

Arithmetic Shifters

Arithmetic shifters are used for shifting binary strings that represent signed integers in two's complement representation. **Since left shifting is the same in logical shifting and in arithmetic shifting, we discuss only right shifting** (i.e., division by a power of 2).

Definition

The binary string $y[n-1:0]$ is an arithmetic right shift by ℓ positions of the binary string $x[n-1:0]$ if the following holds:

$$y[i] \triangleq \begin{cases} x[n-1] & \text{if } i \geq n-\ell \\ x[i+\ell] & \text{if } 0 \leq i < n-\ell. \end{cases}$$

Arithmetic Shifters (cont.)

Example

- $y[3:0] = 0010$ is an arithmetic right shift of $x[3:0] = 0101$ by $\ell = 1$ positions.
- On the other hand, $y[3:0] = 1110$ is an arithmetic right shift of $x[3:0] = 1001$ by $\ell = 2$ positions.
- When we apply an arithmetic right shift by ℓ positions to $x[n-1:0]$, we obtain the string $x[n-1]^{\ell} \circ x[n-1:\ell]$.

Notation.

Let $\mathrm{ARS}(\vec{x}, i)$ denote the arithmetic right shift of $\vec{x}$ by i positions.

An arithmetic right shifter

An arithmetic right shifter is defined as follows.

Definition

An ARITH-SHIFT(n) is a combinational circuit defined as follows:

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

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

Functionality: The output $\vec{y}$ is a (sign-extended) arithmetic right shift of $\vec{x}$ by $\langle \vec{sa} \rangle$ positions. Formally,

$$y[n-1:0] \stackrel{\triangle}{=} \text{ARS}(x[n-1:0], \langle \vec{sa} \rangle).$$

Example

Let $x[3:0] = 1001$. If $sa[1:0] = 10$, then ARITH-SHIFT(4) outputs $y[3:0] = 1110$.

Complete the details!

Compound Adder

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

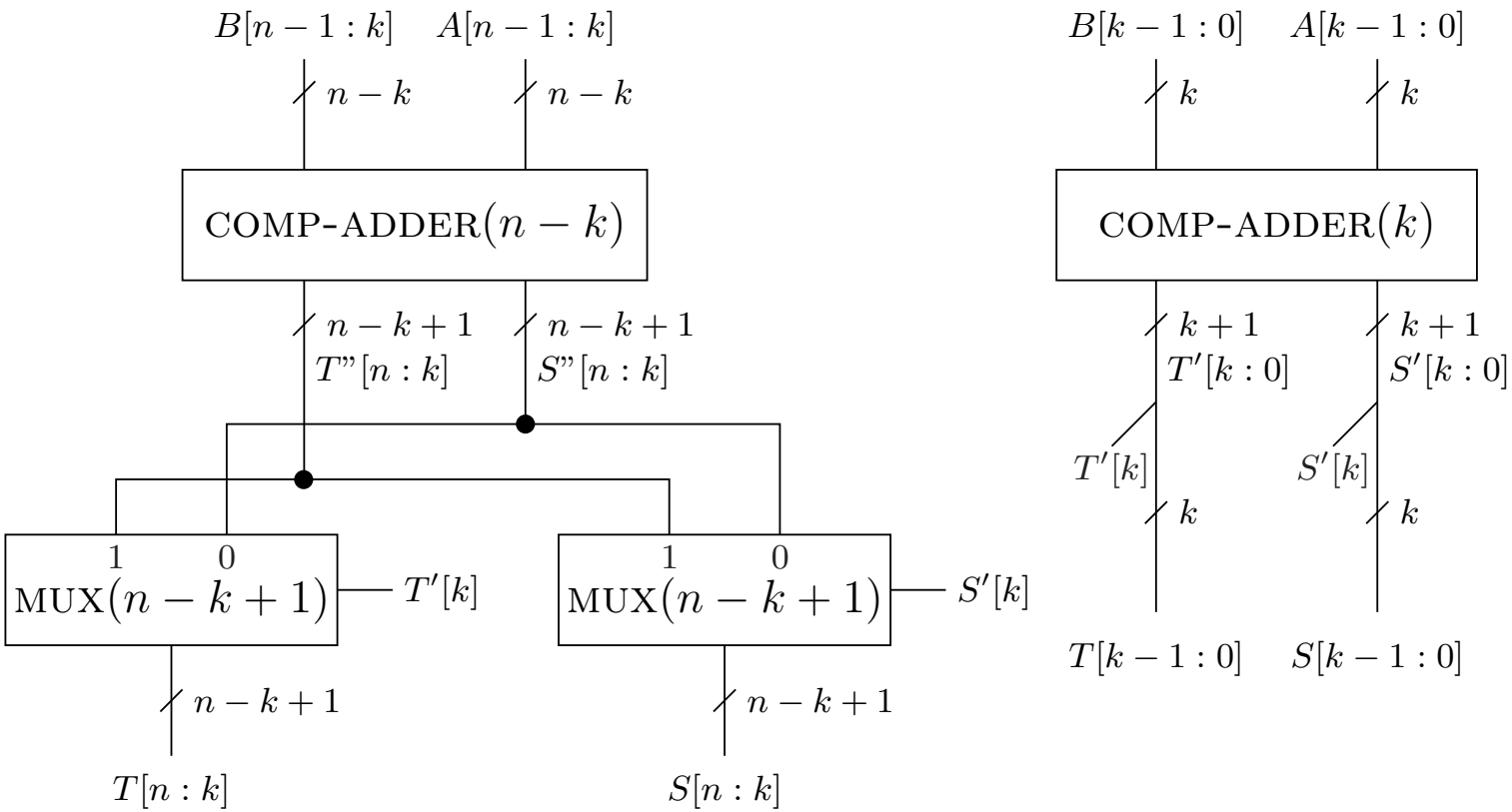
Functionality:

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

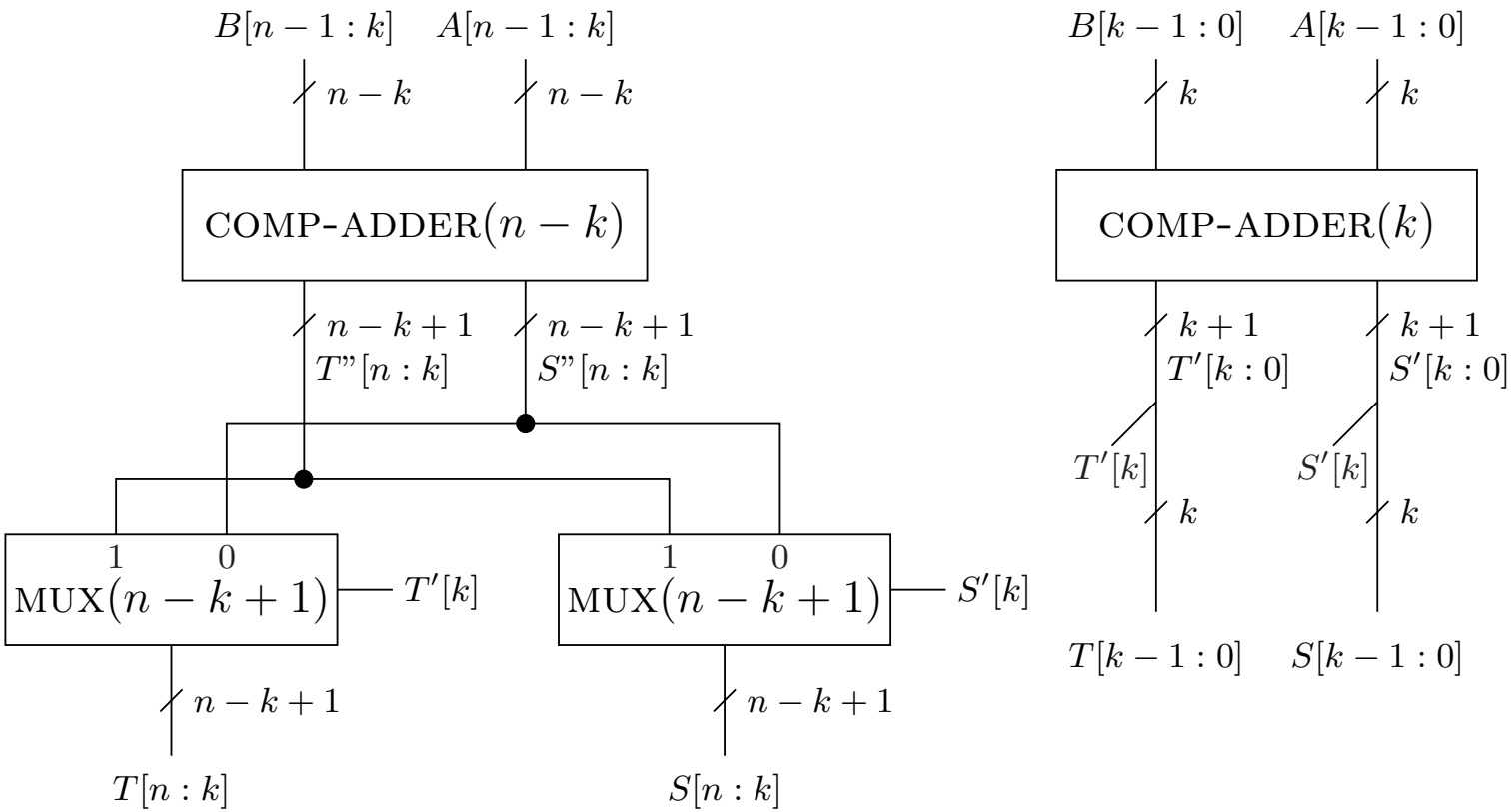
Note that a Compound Adder does not have carry-in input. To simplify notation, the carry-out bits are denoted by $S[n]$ for the sum and by $T[n]$ for the incremented sum.

COMP-ADDER(n) - Implementation

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



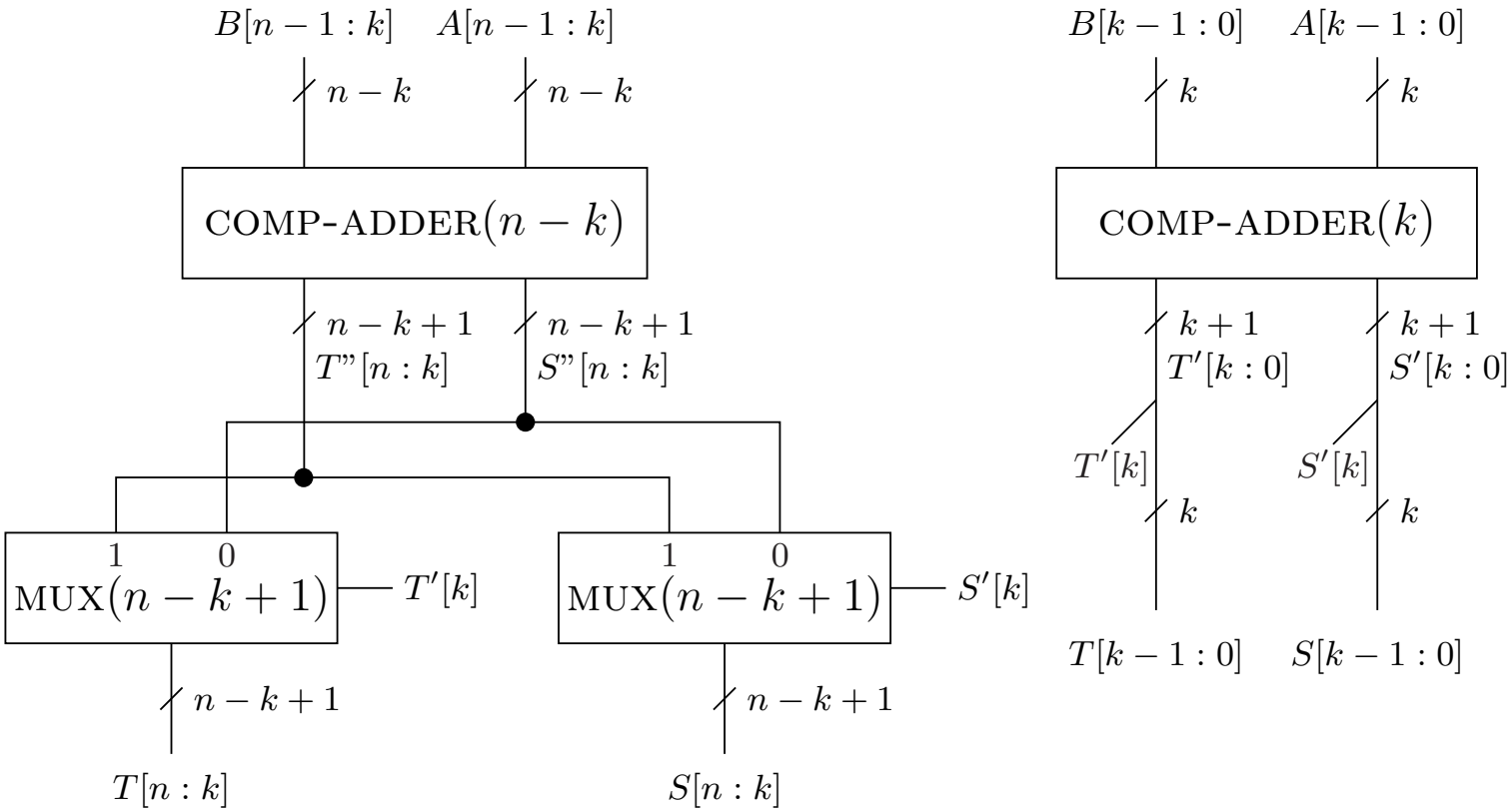
COMP-ADDER(n) - example Implementation



Example

Consider a COMP-ADDER(4) with input $A[3:0] = 0110$ and $B[3:0] = 1001$.

COMP-ADDER(n) - example Implementation



Claim

The COMP-ADDER(n) *design is a correct adder.*

Exam Solution: MOED A2 Fall 2010

Questions 1, 2.