SAARLAND UNIVERSITY
COMPUTER SCIENCE

Prof. Dr. Jan Reineke
Dr. Andreas Abel
Lucas Biehl

Systemarchitektur SS 2021

Präsenzblatt 4 (Lösungsvorschläge)

Hinweis: **Dieses Aufgabenblatt wurde von Tutoren erstellt. Die Aufgaben sind für die Klausur weder relevant noch irrelevant, die Lösungsvorschlage weder korrekt noch inkorrekt.**

Aufgabe 4.1: Geschlossene Form finden

Ordnen Sie den folgenden rekursiven Formeln jeweils alle passenden geschlossenen Formeln zu.

(a) $f(0) = f(1) = 0,$
$f(2k) = 42 + f(2(k-1)), \quad f(2k+1) = f(2k)$

(b) $f(0) = 1, \quad f(n) = 3 \cdot f(n-1)$

(c) $f(0) = 1, \quad f(n) = 7 \cdot f(n-1) + 42$

(d) $f(1) = 1, \quad f(n) = f(\frac{n}{2}) + 2 \quad$ für $n = 2^k$

(e) $f(1) = 2, \quad f(n) = f(\frac{n}{2}) + 2 \quad$ für $n = 2^k$

(i) 3^n

(ii) $42\lfloor \frac{n}{2} \rfloor$

(iii) $8 \cdot 7^n - 7$

(iv) $42\lceil \frac{n}{2} \rceil$

(v) $2(\log_2(n) + 1)$

(vi) $2\log_2(n)$

(vii) $2\log_2(n) + 1$

(viii) $2\log_2(n) - 1$

Finden Sie nun selbst eine geschlossene Form und beweisen Sie die Korrektheit mit vollständiger Induktion.

1. $f(0) = 35, \quad f(n) = 42 + f(n-1)$
2. $f(0) = 10, \quad f(n) = 2 \cdot f(n-1) + 5$

Lösungsvorschlag:

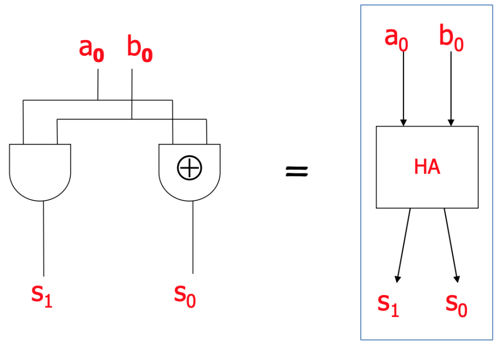
1. a 2; b 1; c 3; d 7; e 5
2. (a) $g(n) = 42n + 35$, denn $f(0) = 35 = g(0)$ und
   $f(n+1) = 42 + f(n) \overset{\text{IV}}{=} 42 + g(n) = 42 + 42n + 35 = 42(n+1) + 35 = g(n+1)$
   (b) $g(n) = 15 \cdot 2^n - 5$, denn $f(0) = 10 = g(0)$ und
   $f(n+1) = 2f(n) + 5 \overset{\text{IV}}{=} 2g(n) + 5 = 2(15 \cdot 2^n - 5) + 5 = 15 \cdot 2^{n+1} - 5 = g(n+1)$

Aufgabe 4.2: Addierer

1. Geben Sie den Schaltkreis eines Halbaddierers an. Was berechnet er?
2. Konstruieren Sie einen Volladdierer aus Halbaddierern. Was ist sein Vorteil gegenüber dem Halbaddierer?
3. Definieren Sie einen rekursiven Schaltkreis für einen Addierer zweier n-Bit Binärzahlen mit linearer Tiefe.

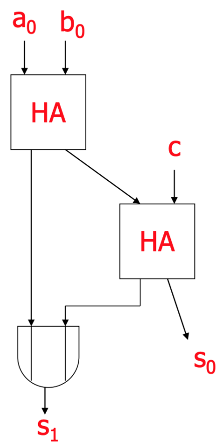
Lösungsvorschlag:

1.



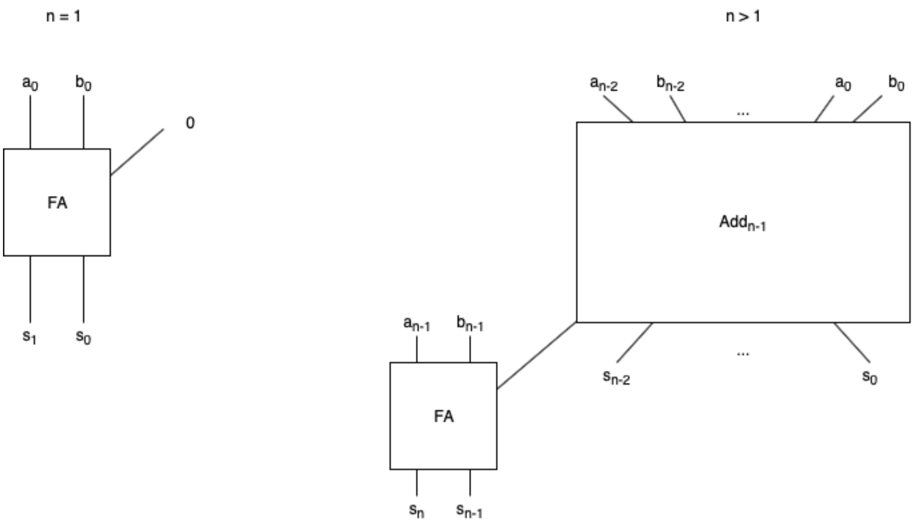
Der Halbaddierer addiert zwei Eingabe-Bits und gibt die Summe als eine zwei-Bit Zahl aus.

2.



Der Volladdierer nimmt zwei Eingabe-Bits und ein zusätzliches Carry Bit und addiert sie zusammen. In dem Carry Bit liegt auch sein entscheidener Vorteil. Es ermöglicht Overflows zu verarbeiten. Dies ist insbesondere wichtig, weil die Addition jeweils Bit-by-Bit durchgeführt wird und daher eventuelle Overflows bei der nächsten Addition beachtet werden können. Der Volladdierer gibt zwei Bits aus. Diese geben als 2-stellige Binärzahl die Anzahl der Einsen der drei Input-Bits an.

3.

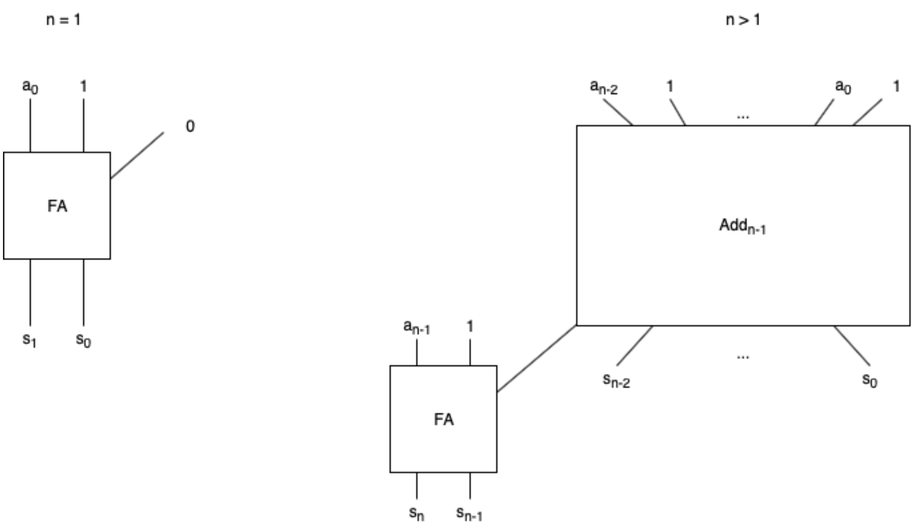


Aufgabe 4.3: Dekrementierer

1. Es sei ein n-Bit Register gegeben, in dem eine Zahl gespeichert ist, die im Zweierkomplement interpretiert wird. Überlegen Sie sich, wie man diese Zahl möglichst einfach dekrementieren kann. Geben Sie eine rekursive Definition eines Schaltkreises an, der diese Aufgabe erfüllt.
2. Bestimmen Sie Kosten und Tiefe rekursiv.
3. Geben Sie die Tiefe in geschlossener Form an und beweisen Sie die Korrektheit.

Lösungsvorschlag:

1. Am einfachsten ist es, n Einsen hinzuzuaddieren, da diese im Zweierkomplement der -1 entsprechen.



2. Kosten und Tiefe in rekursiver Form:
   - $C(Dec_1) = C(FA) = 1 + 2 \cdot C(HA) = 5$
   - $C(Dec_n) = C(FA) + C(Dec_{n-1})$

- $d(Dec_1) = d(FA) = 2 \cdot d(HA) + 1 = 3$
- $d(Dec_n) = 2 + d(Dec_{n-1}) = d(HA) + 1 + d(Dec_{n-1})$

3. Wir zeigen $d(Dec_n) = 3 + 2(n-1)$ durch Induktion über n:

   Induktionsanfang ($n = 1$): $d(Dec_1) = d(FA) = 3 = 3 + 2 \cdot (0) = 3 + 2 \cdot (n-1)$

   Induktionsvoraussetzung: Es gelte $d(Dec_n) = 3 + 2 \cdot (n-1)$ für ein festes $n \in \mathbb{N}$.

   Induktionsschritt ($n \to n+1$):

   $$d(Dec_{n+1}) = d(HA)+1+d(Dec_n) \overset{IV}{=} d(HA)+1+3+2\cdot(n-1) = 2+3+2\cdot(n-1) = 3+2\cdot(n)$$

Aufgabe 4.4: Unequal-Schaltkreise

1. Definieren Sie einen Schaltkreis S_1, welcher zwei n-Bit-Sequenzen miteinander vergleicht, und genau dann 1 ausgibt, wenn die beiden Sequenzen sich in mindestens einer Stelle unterscheiden. Definieren Sie den Schaltkreis rekursiv und mit linearer Tiefe.

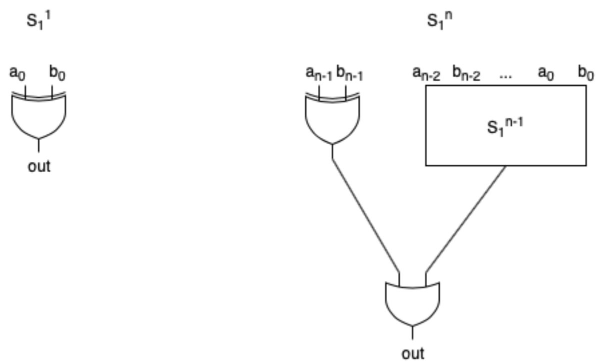
2. Geben Sie einen rekursiven Schaltkreis S_2 an, der die obige Funktion in logarithmischer Tiefe realisiert.

3. Definieren Sie Kosten und Tiefe beider Schaltkreise rekursiv.

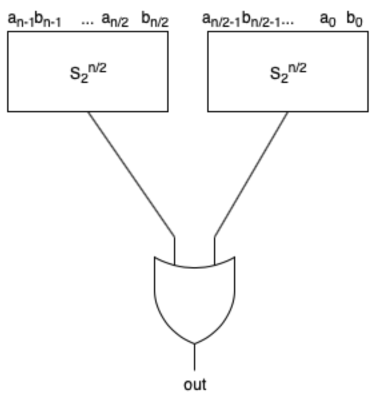
4. Geben Sie die Kosten von S_2 in geschlossener Form an und beweisen Sie die Korrektheit.

Lösungsvorschlag:

1.



2. Wir definieren S_2^1 wie S_1^1 und S_2^n wie folgt:



3. Kosten und Tiefen sind folgendermaßen gegeben:
   - $C_r(S_1^1) = 1$
   - $C_r(S_1^n) = C_r(S_1^{n-1}) + 2$
   - $D_r(S_1) = 1$
   - $D_r(S_1^n) = D_r(S_1^{n-1}) + 1$
   - $C_r(S_2^1) = 1$
   - $C_r(S_2^n) = 2 \cdot C_r(S_2^{\frac{n}{2}}) + 1$
   - $D_r(S_2^1) = 1$
   - $D_r(S_2^n) = D_r(S_2^{\frac{n}{2}}) + 1$
4. Wir zeigen $C(S_2^n) = 2n - 1 = C_r(S_2^n)$ durch Induktions über $n \in \mathbb{N}$:

   Induktionsanfang ($n = 1$): $C(S_2^1) = 2 \cdot 1 - 1 = 1 = C_r(S_2^1)$

   Induktionsvoraussetzung: Es gelte $C(S_2^n) = C_r(S_2^n)$ für ein beliebiges, aber festes $n \in \mathbb{N}$.

   Induktionsschritt ($n \to 2n$):

$$C_r(S_2^{2n}) = 2 \cdot C_r(S_2^n) + 1 \overset{IV}{=} 2 \cdot C(S_2^n) + 1 = 2 \cdot (2n - 1) + 1 = 2(2n) - 2 + 1 = 2(2n) - 1 = C(S_2^{2n})$$

Prof. Dr. Jan Reineke
Dr. Andreas Abel
Lucas Biehl

System Architecture SS 2021

Tutorial Sheet 4 (Suggested Solutions)

***Note:* This task sheet was created by tutors. The tasks are neither relevant nor irrelevant for the exam, the suggested solutions are neither correct nor incorrect.**

Problem 4.1: Finding Closed Forms

Match each of the following recursive formulas with all equivalent closed formulas.

(a) $f(0) = f(1) = 0,$
$f(2k) = 42 + f(2(k-1)), \quad f(2k+1) = f(2k)$

(b) $f(0) = 1, \quad f(n) = 3 \cdot f(n-1)$

(c) $f(0) = 1, \quad f(n) = 7 \cdot f(n-1) + 42$

(d) $f(1) = 1, \quad f(n) = f(\frac{n}{2}) + 2 \quad$ with $n = 2^k$

(e) $f(1) = 2, \quad f(n) = f(\frac{n}{2}) + 2 \quad$ with $n = 2^k$

(i) 3^n

(ii) $42\lfloor \frac{n}{2} \rfloor$

(iii) $8 \cdot 7^n - 7$

(iv) $42\lceil \frac{n}{2} \rceil$

(v) $2(\log_2(n) + 1)$

(vi) $2\log_2(n)$

(vii) $2\log_2(n) + 1$

(viii) $2\log_2(n) - 1$

Now find a closed form and prove the correctness via complete induction.

1. $f(0) = 35, \quad f(n) = 42 + f(n-1)$
2. $f(0) = 10, \quad f(n) = 2 \cdot f(n-1) + 5$

Suggested solution:

1. a 2; b 1; c 3; d 7; e 5
2. (a) $g(n) = 42n + 35$, since $f(0) = 35 = g(0)$ and
   $f(n+1) = 42 + f(n) \overset{\text{IH}}{=} 42 + g(n) = 42 + 42n + 35 = 42(n+1) + 35 = g(n+1)$

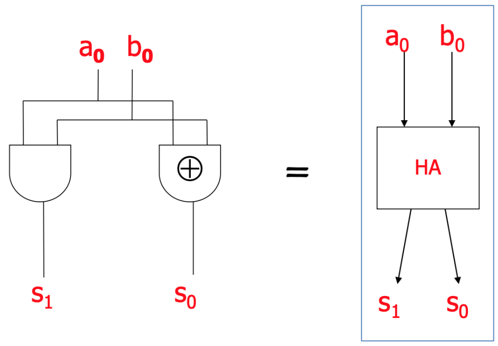
   (b) $g(n) = 15 \cdot 2^n - 5$, since $f(0) = 10 = g(0)$ and
   $f(n+1) = 2f(n) + 5 \overset{\text{IH}}{=} 2g(n) + 5 = 2(15 \cdot 2^n - 5) + 5 = 15 \cdot 2^{n+1} - 5 = g(n+1)$

Problem 4.2: Adders

1. Draw a circuit of a half adder. What does it calculate?
2. Construct a full adder using half adders. What is its advantage over the half adder?
3. Now define a recursive circuit for an adder of two n-bit binary numbers with linear depth.

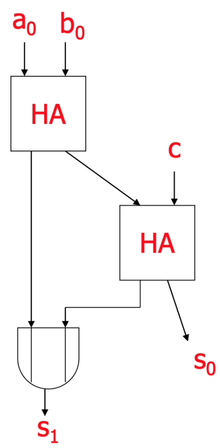
Suggested solution:

1.



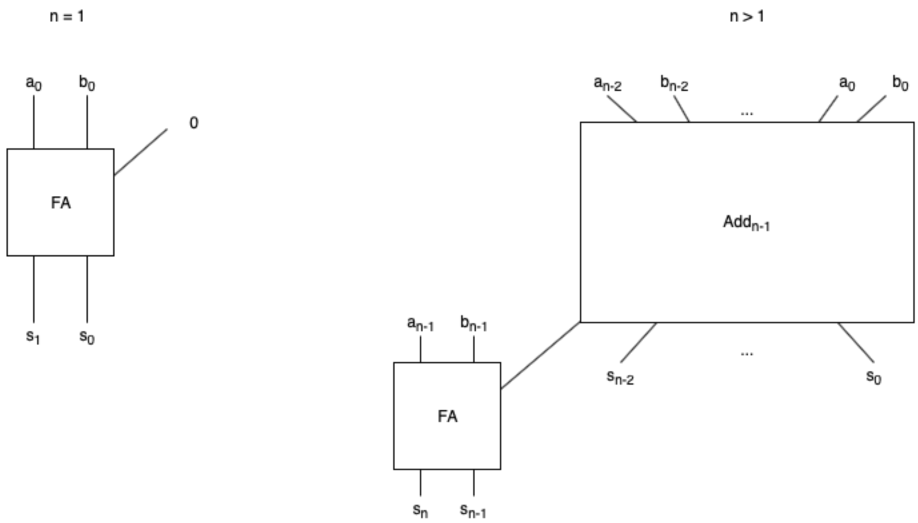
The half adder adds two input bits and outputs the sum as a two-bit number.

2.



The full adder takes two input bits and a carry bit and adds them together. The carry bit is also its decisive advantage. It makes it possible to process overflows. This is especially important because the addition is performed bit-by-bit so that any overflows can be taken into account in the next addition. The full adder outputs two bits indicating the number of ones of the three input bits as a 2-digit binary number.

3.

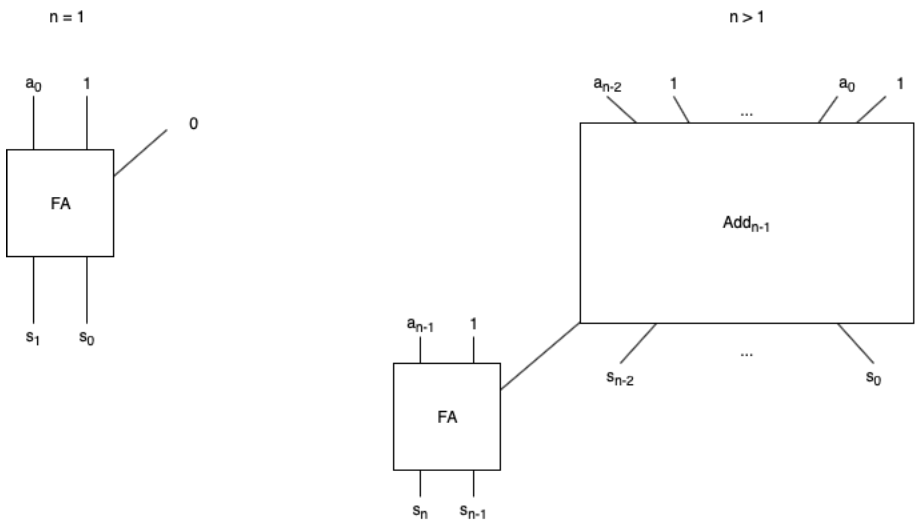


Problem 4.3: Decrementer

1. Let there be an n-bit register storing a number interpreted in two's complement. Consider how to decrement this number as easily as possible. Give a recursive definition of a circuit that accomplishes this task.
2. Determine cost and depth recursively.
3. Find a closed form for the depth and prove it correct.

Suggested solution:

1. The easiest way is to add n ones, because they correspond to -1 in two's complement.



2. Cost and depth in recursive form:
   - $C(Dec_1) = C(FA) = 1 + 2 \cdot C(HA) = 5$
   - $C(Dec_n) = C(FA) + C(Dec_{n-1})$
   - $d(Dec_1) = d(FA) = 2 \cdot d(HA) + 1 = 3$

- $d(Dec_n) = 2 + d(Dec_{n-1}) = d(HA) + 1 + d(Dec_{n-1})$

3. We prove $d(Dec_n) = 3 + 2(n-1)$ by complete induction over $n \in \mathbb{N}$:

   Base case ($n = 1$): $d(Dec_1) = d(FA) = 3 = 3 + 2 \cdot (0) = 3 + 2 \cdot (n-1)$

   Induction hypothesis: Assume $d(Dec_n) = 3 + 2 \cdot (n-1)$ for a fixed $n \in \mathbb{N}$.

   Induction step ($n \to n+1$):

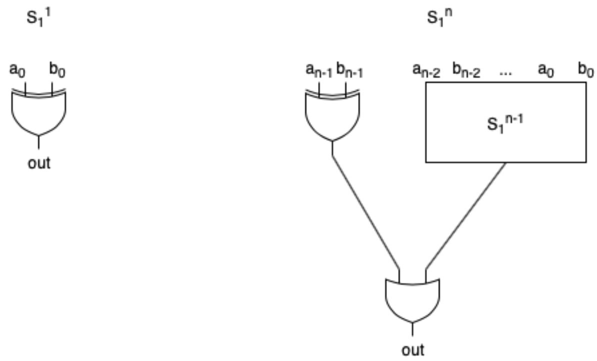
$$d(Dec_{n+1}) = d(HA)+1+d(Dec_n) \overset{\text{IH}}{=} d(HA)+1+3+2\cdot(n-1) = 2+3+2\cdot(n-1) = 3+2\cdot(n)$$

Problem 4.4: Unequal Circuits

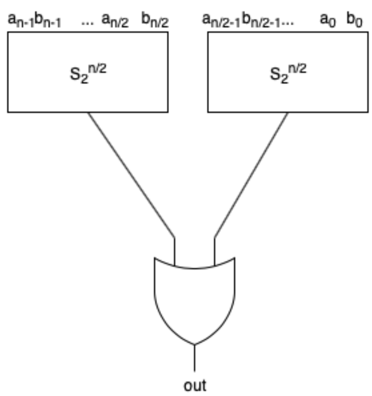
1. Define a circuit S_1 that compares two n-bit sequences with each other and outputs 1 iff the two sequences differ in at least one place. Define the circuit recursively and with linear depth.
2. Can you also specify a recursive circuit S_2 realizing the above function in logarithmic depth?
3. Define cost and depth of both circuits recursively.
4. Give the cost of S_2 in closed form and prove it correct.

Suggested solution:

1.



2. We define S_2^1 to be S_1^1 and S_2^n as follows:



3. Cost and depth are given as follows:
   - $C_r(S_1^1) = 1$

- $C_r(S_1^n) = C_r(S_1^{n-1}) + 2$
- $D_r(S_1) = 1$
- $D_r(S_1^n) = D_r(S_1^{n-1}) + 1$
- $C_r(S_2^1) = 1$
- $C_r(S_2^n) = 2 \cdot C_r(S_2^{\frac{n}{2}}) + 1$
- $D_r(S_2^1) = 1$
- $D_r(S_2^n) = D_r(S_2^{\frac{n}{2}}) + 1$

4. We show $C(S_2^n) = 2n - 1 = C_r(S_2^n)$ by complete induction over $n \in \mathbb{N}$:

   Base case ($n = 1$): $C(S_2^1) = 2 \cdot 1 - 1 = 1 = C_r(S_2^1)$

   Induction hypothesis: Assume $C(S_2^n) = C_r(S_2^n)$ for a fixed $n \in \mathbb{N}$.

   Inductions step ($n \to 2n$):

$$C_r(S_2^{2n}) = 2 \cdot C_r(S_2^n) + 1 \overset{\text{IH}}{=} 2 \cdot C(S_2^n) + 1 = 2 \cdot (2n-1) + 1 = 2(2n) - 2 + 1 = 2(2n) - 1 = C(S_2^{2n})$$