Die folgenden Folien behandeln die Herleitung eines Carry Lookahead Addierers mit linearen Kosten und logarithmischer Tiefe.

Eine entsprechende Abhandlung ist auch im Buch von Keller / Paul (Hardwaredesign), S. 194-198 zu finden.

Der Stoff ist **nicht** klausurrelevant. Ich wurde nur gefragt, wie ein CLA funktioniert, im Folgenden ist die Antwort zu finden.

15.12.2015

Christoph Scholl

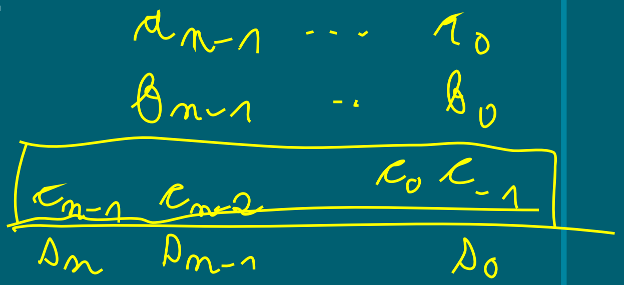
Der Carry Lookahead Addierer (CLA)

Gibt es Addierer mit linearen Kosten und logarithmischer Tiefe? → Carry Lookahead Addierer

Der Carry Lookahead Addierer (CLA)

Gibt es Addierer mit linearen Kosten und logarithmischer Tiefe? → Carry Lookahead Addierer

Übertragbit



Idee:

Man kann das Problem reduzieren auf die schnelle Berechnung des Übertragbits c_i. Sind c_i bekannt, so ergibt sich s_i durch $a_i \oplus b_i \oplus c_{i-1}$.

$s_i = a_i \oplus b_i \oplus \boxed{c_{i-1}}$

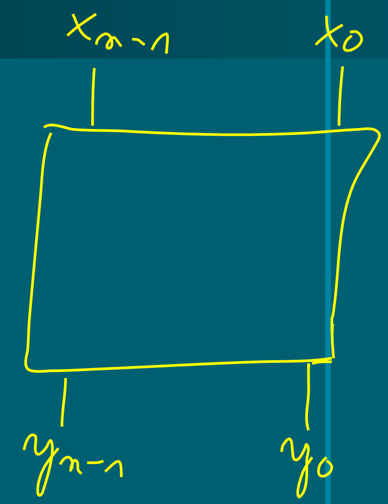
Berechnung der c_i durch parallele Präfix-Berechnung!

Parallele Präfix-Berechnung

Sei M eine Menge,

$\circ : M \times M \rightarrow M$ eine assoziative Abbildung.

Parallele Präfix-Berechnung



Sei M eine Menge,

$\circ : M \times M \to M$ eine assoziative Abbildung.

Problem:

Realisiere die parallele Präfix-Funktion $PP_\circ^n$ durch

$$PP_\circ^n : M^n \to M^n, \quad PP_\circ^n(x_{n-1},\ldots,x_0) = (y_{n-1},\ldots,y_0) \quad \text{mit}$$

$$y_i = x_i \circ \ldots \circ x_0 \qquad \text{für } 0 \le i \le n.$$

$y_0 = x_0$

$y_1 = x_1 \circ x_0$

$y_2 = x_2 \circ x_1 \circ x_0$

$\vdots$

$y_{n-1} = x_{n-1} \circ \ldots \circ x_2 \circ x_1 \circ x_0$

Parallele Präfix-Berechnung (ff)

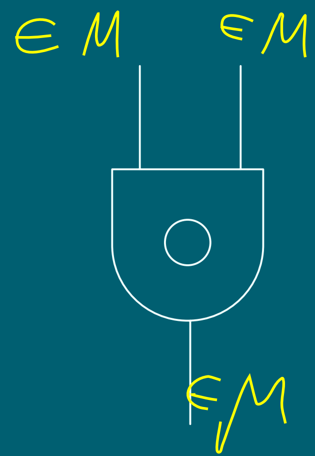
Annahme:

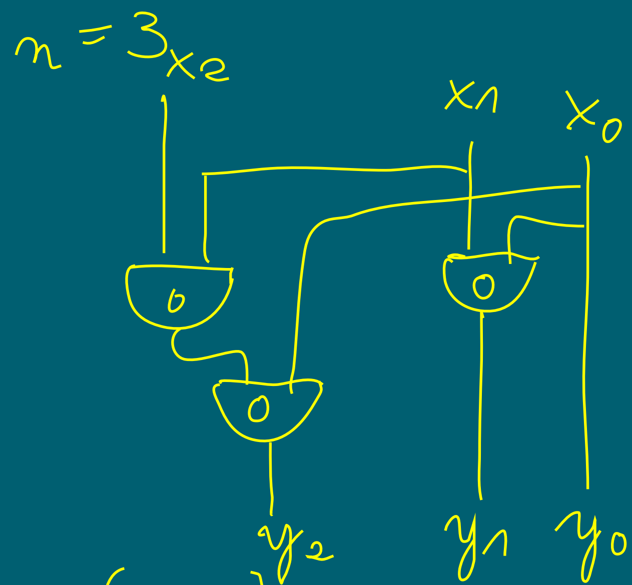
- wird durch ein spezielles Gatter berechnet:

Parallele Präfix-Berechnung (ff)

<u>Annahme</u>:

- wird durch ein spezielles Gatter berechnet:





$= (x_2 \circ x_1) \circ x_0 \qquad = x_1 \circ x_0$

$n = 4: \quad y_3 = ((x_3 \circ x_2) \circ (x_1 \circ x_0))$

Vorschläge zur Realisierung:

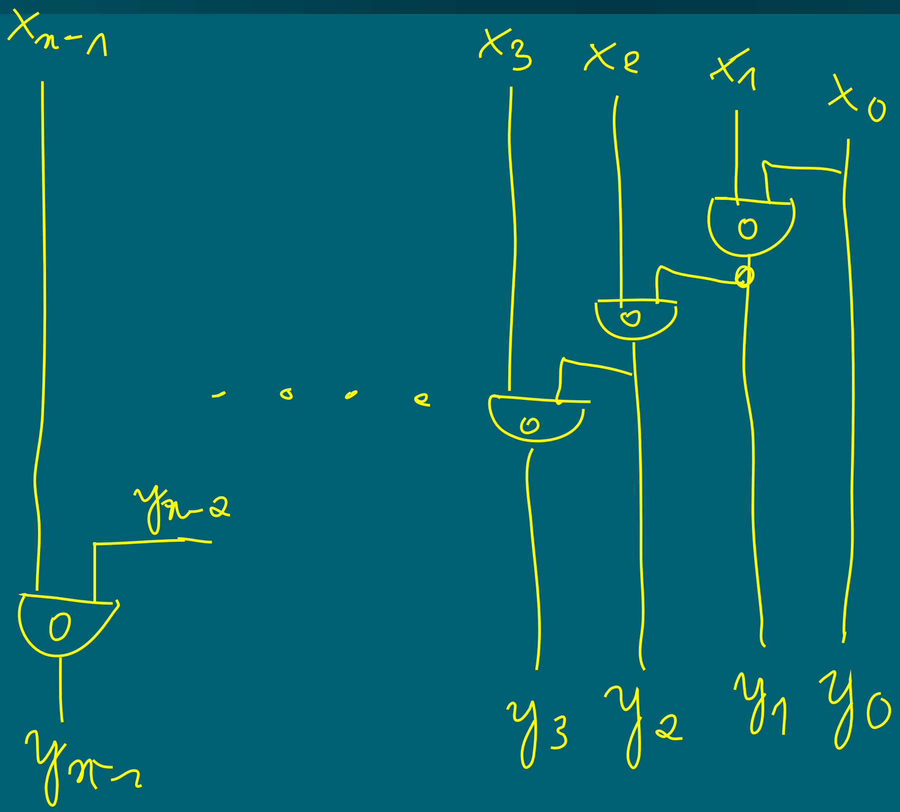
1. Vorschlag: Realisiere alle y_i getrennt durch balancierte Bäume

$\Longrightarrow$ Kosten $O(n^2)$, Tiefe $O(\log(n))$

2. Vorschlag:

$$y_0 = x_0$$
$$y_1 = x_1 \circ y_0$$
$$y_2 = x_2 \quad y_1$$
$$\vdots$$
$$y_{n-1} = x_{n-1} \circ y_{n-2}$$

$\Longrightarrow$ Kosten $O(n)$, Tiefe $O(n)$


x_{n-1}
x_3
x_2
x_1
x_0
y_{n-2}
y_{n-1}
y_3
y_2
y_1
y_0

$$y_0 = x_0$$

$$\vdots$$

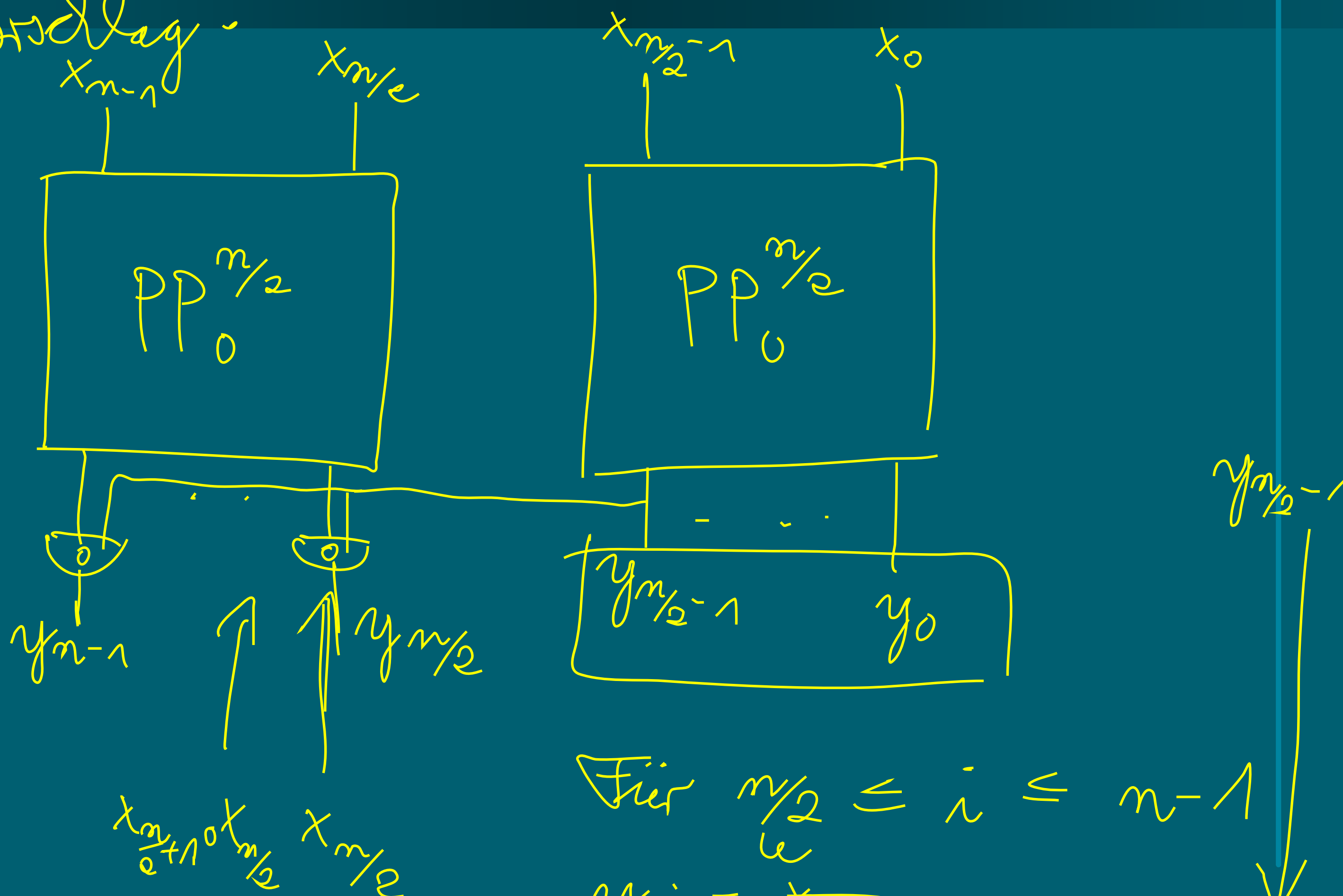
$$y_{n/2-1} = \underline{\underline{(x_{n/2-1} \circ \dots \circ x_0)}}$$

$$y_{n/2} = x_{n/2} \circ \underline{\underline{y_{n/2-1}}}$$

$$\vdots$$

$$y_{n-1} = x_{n-1} \circ \dots \circ x_{n/2} \circ \underline{y_{n/2-1}}$$

3. Vorschlag:



Für $n/2 \leq i \leq n-1$

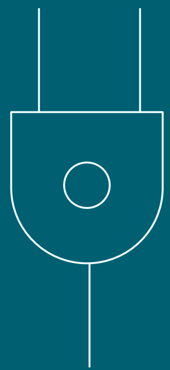
$y_i = \cancel{x_{i-1} \circ} \; x_i \circ \ldots \circ x_{n/2} \circ y_{n/2-1}$

$\Rightarrow$ Tiefe $O(\log n)$, Kosten $O(n \log n)$

Parallele Präfix-Berechnung (ff)

Annahme:

$\circ$ wird durch ein spezielles Gatter berechnet:



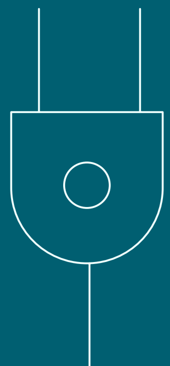
Idee:

Nutze Assoziativität von $\circ$!

Parallele Präfix-Berechnung (ff)

Annahme:

$\circ$ wird durch ein spezielles Gatter berechnet:



Beispiel:

$\oplus$ - Funktion für $\circ$.

$\circ$ ist assoziativ.

Idee:

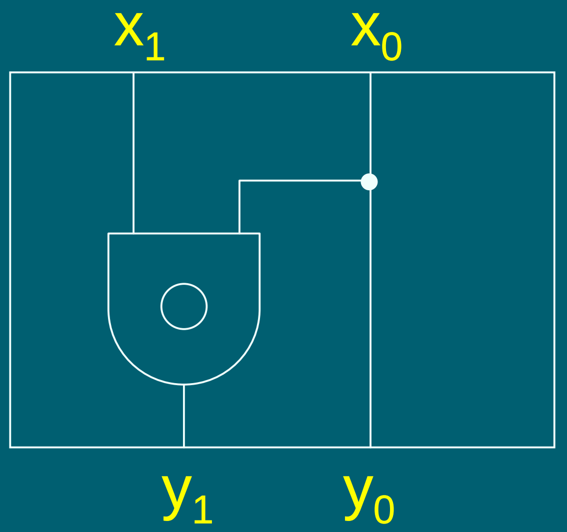
Nutze Assoziativität von $\circ$!

Grundgatter für parallele Präfixberechnung (n=2)

Realisierung von $PP_{\circ}^{2}$:

Grundgatter für parallele Präfixberechnung (n=2)

Realisierung von $PP_{\circ}^{2}$:



Funktionsweise der parallelen Präfixberechnung

Sei $n = 2^k$.

Funktionsweise der parallelen Präfixberechnung

Sei $n = 2^k$.

$$\begin{aligned} y_{2i+1} &= x_{2i+1} \circ x_{2i} \circ \ldots \circ x_1 \circ x_0 \\ &= \left(x_{2i+1} \circ x_{2i}\right) \circ \ldots \circ \left(x_1 \circ x_0\right) \qquad \left(\forall i \in \left\{0, \ldots, \tfrac{n}{2} - 1\right\}\right) \end{aligned}$$

Funktionsweise der parallelen Präfixberechnung

Sei n = 2^k.

$$y_{2i+1} = x_{2i+1} \circ x_{2i} \circ \ldots \circ x_1 \circ x_0$$

$$= \underbrace{\left(x_{2i+1} \circ x_{2i}\right)}_{x'_i} \circ \ldots \circ \underbrace{\left(x_1 \circ x_0\right)}_{x'_0} \qquad \left(\forall i \in \left\{0, \ldots, \tfrac{n}{2} - 1\right\}\right)$$

$$\underline{y_{2i}} = x_{2i} \circ x_{2i-1} \circ x_{2i-2} \circ \ldots \circ x_1 \circ x_0$$

$$= \underline{x_{2i}} \circ \underbrace{\left(x_{2i-1} \circ x_{2i-2}\right)}_{x'_{i-1}} \circ \ldots \circ \underbrace{\left(x_1 \circ x_0\right)}_{x'_0} \qquad \left(\forall i \in \left\{0, \ldots, \tfrac{n}{2} - 1\right\}\right)$$

Funktionsweise der parallelen Präfixberechnung (ff)

Mit $x_i' := x_{2i+1} \circ x_{2i}$ gilt also:

Funktionsweise der parallelen Präfixberechnung (ff)

Mit $x_i' := x_{2i+1} \circ x_{2i}$ gilt also:

$$y_{2i+1} = x'_i \circ \ldots \circ x'_0 := \underline{y'_i} \qquad \left(\forall i \in \left\{0,\ldots,\tfrac{n}{2}-1\right\}\right)$$

Funktionsweise der parallelen Präfixberechnung (ff)

Mit $x_i' := x_{2i+1} \circ x_{2i}$ gilt also:

$$y_{2i+1} = x'_i \circ \ldots \circ x'_0 := y'_i \qquad \left(\forall i \in \left\{0, \ldots, \tfrac{n}{2} - 1\right\}\right)$$

$$\begin{aligned} y_{2i} &= x_{2i} \circ x'_{i-1} \circ \ldots \circ x_0' \qquad \left(\forall i \in \left\{0, \ldots, \tfrac{n}{2} - 1\right\}\right) \\ &= x_{2i} \circ y_{2i-1} = x_{2i} \circ y'_{i-1} \end{aligned}$$

Funktionsweise der parallelen Präfixberechnung (ff)

Mit $x_i' := x_{2i+1} \circ x_{2i}$ gilt also:

$$y_{2i+1} = x'_i \circ \ldots \circ x'_0 := y'_i \qquad \left(\forall i \in \left\{0,\ldots,\tfrac{n}{2}-1\right\}\right)$$

$$y_{2i} = x_{2i} \circ x'_{i-1} \circ \ldots \circ x_0' \qquad \left(\forall i \in \left\{0,\ldots,\tfrac{n}{2}-1\right\}\right)$$

$$= x_{2i} \circ y_{2i-1} = x_{2i} \circ y'_{i-1}$$

$$\vdots$$

$$y_0 = x_0$$

Vereinfachte Beschreibung des Vorgehens

1.Schritt:

Fasse jeweils benachbarte Paare x_{2i+1}, x_{2i} zusammen:

Vereinfachte Beschreibung des Vorgehens

1.Schritt:

Fasse jeweils benachbarte Paare x_{2i+1}, x_{2i} zusammen:

$$x'_i = x_{2i+1} \circ x_{2i}$$

Vereinfachte Beschreibung des Vorgehens (ff)

2.Schritt:

Benutze Schaltkreis $P_{n/2}$ mit Inputs x'_i , $0 \leq i \leq n/2 - 1$:

Vereinfachte Beschreibung des Vorgehens (ff)

<u>2.Schritt</u>:

Benutze Schaltkreis $P_{n/2}$ mit Inputs x'_i , $0 \leq i \leq n/2 - 1$:

$$
\begin{aligned}
y'_0 &= x'_0 &&= x_1 \circ x_0 &&= y_1 \\
y'_1 &= x'_1 \circ x'_0 &&= (x_3 \circ x_2) \circ (x_1 \circ x_0) &&= y_3 \\
y'_2 &= x'_2 \circ x'_1 \circ x'_0 &&= (x_5 \circ x_4) \circ (x_3 \circ x_2) \circ (x_1 \circ x_0) &&= y_5 \\
&\vdots && \vdots && \vdots \\
y'_{\frac{n}{2}-1} &= x'_{\frac{n}{2}-1} \circ \ldots \circ x'_0 &&= (x_{n-1} \circ x_{n-2}) \circ \ldots \circ (x_1 \circ x_0) &&= y_{n-1}
\end{aligned}
$$

Vereinfachte Beschreibung des Vorgehens (ff)

3.Schritt:

Ergänze die fehlenden y_i mit i gerade:

Vereinfachte Beschreibung des Vorgehens (ff)

<u>3.Schritt</u>:

Ergänze die fehlenden y_i mit i gerade:

$$y_{2i} = \boxed{x_{2i}} \circ \underbrace{\left(x_{2i-1} \circ \ldots \circ x_0\right)} = x_{2i} \circ y_{2i-1} \qquad 1 \le i \le \tfrac{n}{2} - 1$$

$$y_0 = x_0$$

Vereinfachte Beschreibung des Vorgehens (ff)

3.Schritt:

Ergänze die fehlenden y_i mit i gerade:

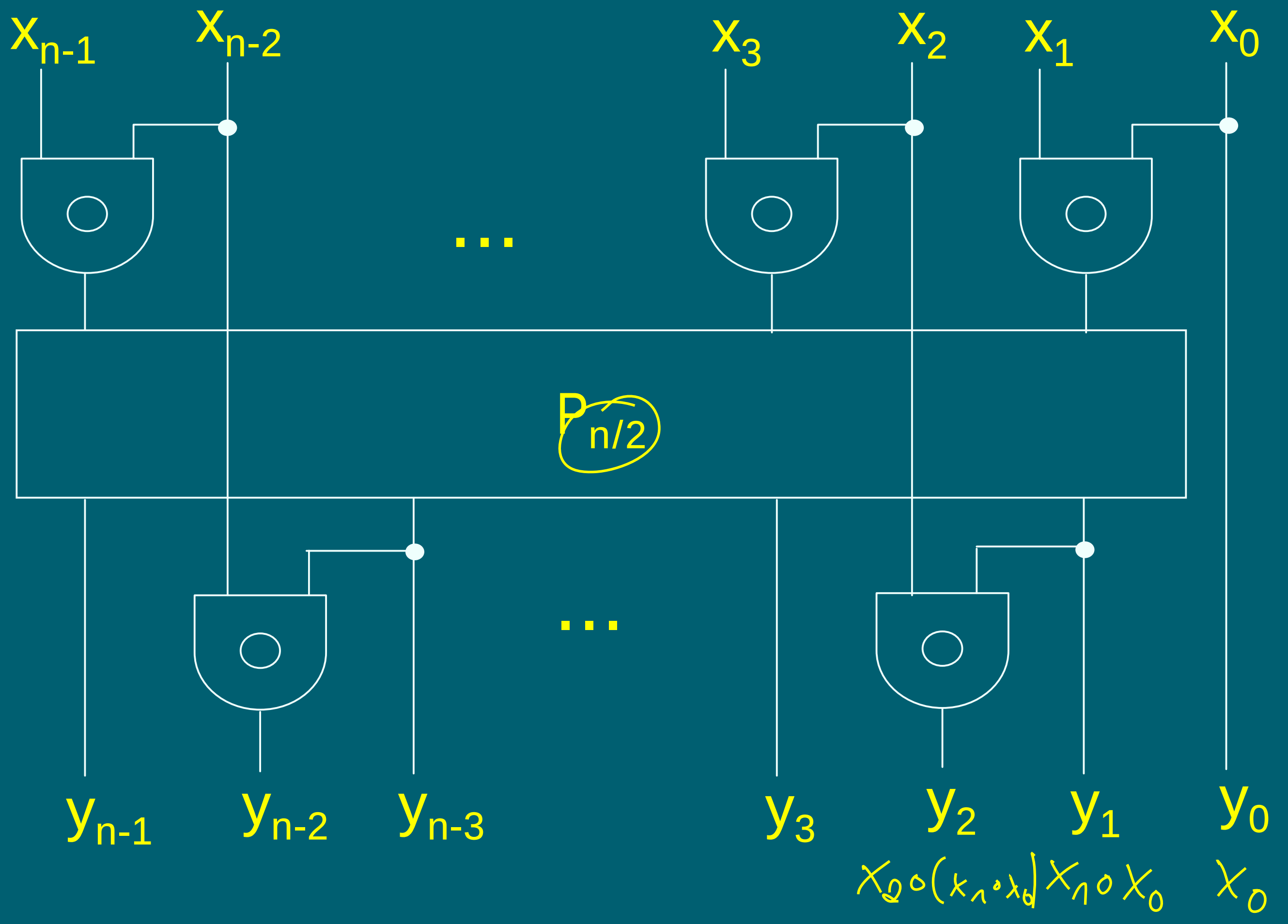
$$y_{2i} = x_{2i} \circ \left(x_{2i-1} \circ \ldots \circ x_0\right) = x_{2i} \circ y_{2i-1} \qquad 1 \le i \le \tfrac{n}{2} - 1$$

$$y_0 = x_0$$

→ Schaltkreis P_n zur Realisierung von $PP_\circ^n$

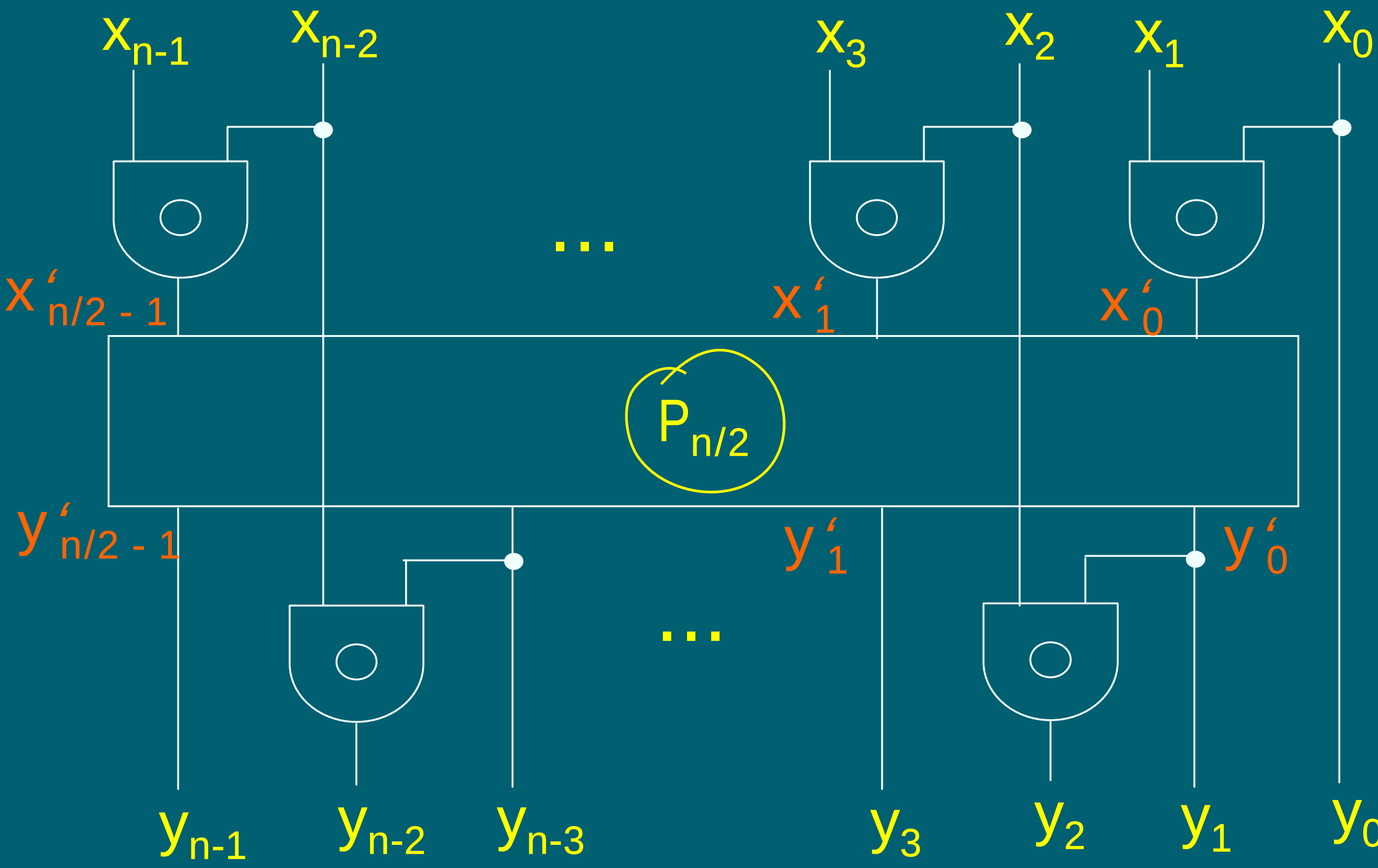
Schaltkreis P_n

Schaltkreis P_n



Schaltkreis P_n

$n = 2^k$



2^{k-1} Gatter

$2^{k-1} - 1$ Gatter

Lemma 10.2

Lemma 10.2

P_n hat Kosten $C(P_n) \leq 2n$ und

Tiefe depth $(P_n) \leq 2 \log(n) - 1$ für $n=2^k$.

Lemma 10.2

P_n hat Kosten $C(P_n) \leq 2n$ und

Tiefe depth $(P_n) \leq 2\log(n) - 1$ für $n=2^k$.

Beweis:

$n=2^k$, also $C(P_n) \leq 2^{k+1}$, $\text{depth}(P_n) \leq 2k-1$

Lemma 10.2

PP_0^n P_n

P_n hat Kosten $C(P_n) \le 2n$ und

Tiefe depth $(P_n) \le 2\log(n) - 1$ für $n=2^k$.

<u>Beweis</u>:

$n=2^k$, also $C(P_n) \le 2^{k+1}$, $\text{depth}(P_n) \le 2k-1$

1) Kosten:

P_2 $P_{2^{k-1}}$ „Zusatzgatter"

k=1: $C(\cancel{P_0^2}) = 1$

k-1 $\to$ k: $C(\cancel{P_0^{2^k}}) \le C(\cancel{P_0^{2^{k-1}}}) + \underbrace{2^{k-1} \cdot 2}$

P_{2^k} $\le 2^k + 2^k = 2^{k+1}$

Lemma 10.2 (ff)

Beweis:

2) Tiefe:

P_2

k=1: $\quad \mathrm{depth}(\cancel{P_0^2}) = 1$

$P_{2^{k-1}}$

k-1 $\rightarrow$ k: $\quad \mathrm{depth}(\cancel{P_0^{2^k}}) = 2 + \mathrm{depth}(\cancel{P_0^{2^{k-1}}})$

P_{2^k}

$$\le 2 + (2(k-1) - 1)$$

$$= \underline{2k - 1}$$

Konstruktion des Carry Lookahead Addierers

Für $0 \leq i < n$: $s_i = a_i \oplus b_i \oplus c_{i-1}$

→ Schnelle Berechnung der c_{i-1} genügt

Konstruktion des Carry Lookahead Addierers

Für $0 \leq i < n$: $s_i = a_i \oplus b_i \oplus c_{i-1}$

→ Schnelle Berechnung der c_{i-1} genügt

$$a_{i-1} \ldots a_0$$
$$b_{i-1} \ldots b_0$$
$$c$$

Konstruktion des Carry Lookahead Addierers

Für $0 \leq i < n$: $s_i = a_i \oplus b_i \oplus c_{i-1}$

→ Schnelle Berechnung der c_{i-1} genügt

$$\begin{array}{r} a_{i-1} \ldots a_0 \\ b_{i-1} \ldots b_0 \\ c \end{array}$$

Konstruktion des Carry Lookahead Addierers

Für $0 \leq i < n$: $s_i = a_i \oplus b_i \oplus c_{i-1}$

→ Schnelle Berechnung der c_{i-1} genügt

$$
\begin{array}{l|l}
a_j \ldots a_i & a_{i-1} \ldots a_0 \\
b_j \ldots b_i & b_{i-1} \ldots b_0 \\
\quad c_{i-1} & \qquad\quad c
\end{array}
$$

Konstruktion des Carry Lookahead Addierers

Für $0 \leq i < n$: $s_i = a_i \oplus b_i \oplus c_{i-1}$

→ Schnelle Berechnung der c_{i-1} genügt

$$a_j \ldots a_i \quad a_{i-1} \ldots a_0$$

$$b_j \ldots b_i \quad b_{i-1} \ldots b_0$$

$$c_{i-1} \qquad c$$

Konstruktion des Carry Lookahead Addierers

Für $0 \leq i < n$: $\quad s_i = a_i \oplus b_i \oplus c_{i-1}$

→ Schnelle Berechnung der c_{i-1} genügt

$$\begin{array}{rccc} & & \overbrace{}^{j \qquad i} & \\ & a_{n-1}\dots a_{j+1} & a_j \dots a_i & a_{i-1}\dots a_0 \\ + & b_{n-1}\dots b_{j+1} & b_j \dots b_i & b_{i-1}\dots b_0 \\ + & c_j & c_{i-1} & c \end{array}$$

Frage: Analysiere für gegebene $a_j \dots a_i$, $b_j \dots b_i$ das Verhalten von c_j in Abhängigkeit von c_{i-1}.

Bsp.:

1)

$$e_j \quad \begin{array}{c|cccccc|}
 & j & & & & & i \\
 & 1 & 0 & \cdots & 0 & 0 & \\
 & 1 & 0 & \cdots & 0 & 0 & \\
1 & 0 & 0 & & 0 & 0 & c_{i-1} \\
\hline
 & 0 & 0 & \cdots & & 0 & c_{i-1}
\end{array} \qquad e_j = 1$$

2)

$$e_j \quad \begin{array}{c|cccccc|}
 & j & & & & & i \\
 & 1 & 1 & \cdots & & 1 & 1 \\
 & 0 & 0 & \cdots & & 0 & 0 \\
c_{i-1} & c_{i-1} & c_{i-1} & & c_{i-1} & c_{i-1} & c_{i-1} \\
\hline
 & \overline{c_{i-1}} & \overline{c_{i-1}} & & & \overline{c_{i-1}} & \overline{c_{i-1}}
\end{array} \qquad e_j = c_{i-1}$$

3)

$$e_j \quad \begin{array}{c|cccccc|}
 & 0 & 0 & \cdots & & 0 & 0 \\
 & 0 & 0 & \cdots & & 0 & 0 \\
0 & 0 & 0 & & 0 & 0 & c_{i-1} \\
\hline
 & 0 & 0 & & & 0 & c_{i-1}
\end{array} \qquad e_j = 0$$

Schnelle Berechnung der c_{i-1}

Betrachte Stellen i bis j, $i \leq j$.

Für Belegungen von $(a_j \dots a_i)$ und $(b_j \dots b_i)$ können genau 3 Fälle auftreten:

1) $c_j = 1$ unabhängig von c_{i-1}, d.h. für $c_{i-1} = 0$ und $c_{i-1} = 1$

   Sprechweise:

   Stellen i bis j generieren einen Übertrag.

Schnelle Berechnung der c_{i-1} (ff)

2) $c_j = 1 \Leftrightarrow c_{i-1} = 1$ $\qquad c_{i-1} = c_j$

Sprechweise:

Stellen i bis j propagieren einen Übertrag.

3) $c_j = 0$ unabhängig von c_{i-1}, d.h. für $c_{i-1} = 0$ und $c_{i-1} = 1$

Sprechweise:

Stellen i bis j eliminieren einen Übertrag.

c_{i-1}	generieren c_j	propagieren c_j	eliminieren c_j
0	1	0	0
1	1	1	0

Funktionsdefinition

Definiere die Funktion

$g_{j,i}$, $p_{j,i} : \{0,1\}^{2n} \to \{0,1\}$ für $0 \leq i \leq j < n$ mit

$$g_{j,i}(a,b) = \begin{cases} 1: & \text{Stellen i bis j generieren Übertrag} \\ 0: & \text{sonst} \end{cases}$$

$$p_{j,i}(a,b) = \begin{cases} 1: & \text{Stellen i bis j propagieren Übertrag} \\ 0: & \text{sonst} \end{cases}$$

Funktionsdefinition

Definiere die Funktion

B_{2n}

$g_{j,i}$, $p_{j,i}$: $\{0,1\}^{2n} \rightarrow \{0,1\}$ für $0 \leq i \leq j < n$ mit

$$g_{j,i}(a,b) = \begin{cases} 1: & \text{Stellen } i \text{ bis } j \text{ generieren Übertrag} \\ 0: & \text{sonst} \end{cases}$$

$$p_{j,i}(a,b) = \begin{cases} 1: & \text{Stellen } i \text{ bis } j \text{ propagieren Übertrag} \\ 0: & \text{sonst} \end{cases}$$

$$\begin{array}{l|l|l} a_{n-1} \; \dots & a_j \; \dots \; a_i & \dots \; a_0 \\ b_{n-1} \; \dots & b_j \; \dots \; b_i & \dots \; b_0 \end{array}$$

<u>Bemerkung</u>:

$g_{j,i}$ und $p_{j,i}$ hängen nur ab von $a_j, \dots, a_i, b_j, \dots, b_i$!

Eigenschaften von $g_{j,i}$, $p_{j,i}$

$= a_i \cdot \overline{b_i} + \overline{a_i} \cdot b_i =$

1) $p_{i,i} = a_i \oplus b_i$ für $0 \leq i < n$

   $g_{i,i} = a_i \wedge b_i$ für $0 \leq i < n$

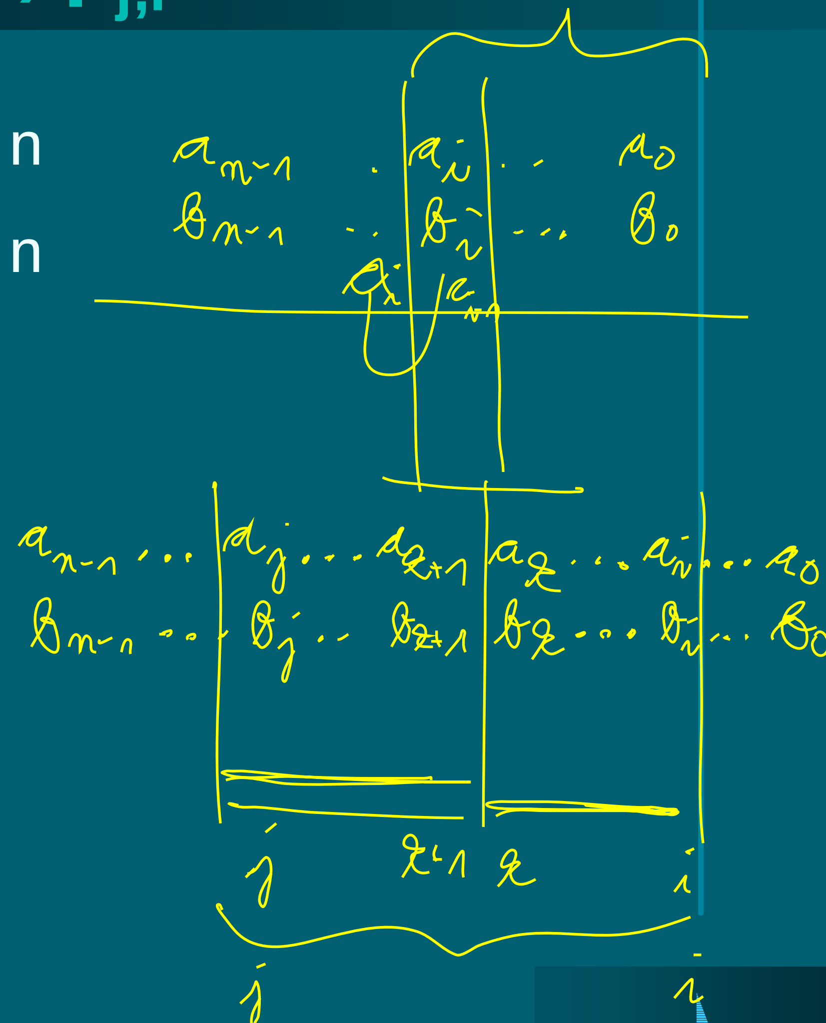
2) Für $i \leq k < j$:

   $g_{j,i} = g_{j,k+1} \vee (g_{k,i} \wedge p_{j,k+1})$

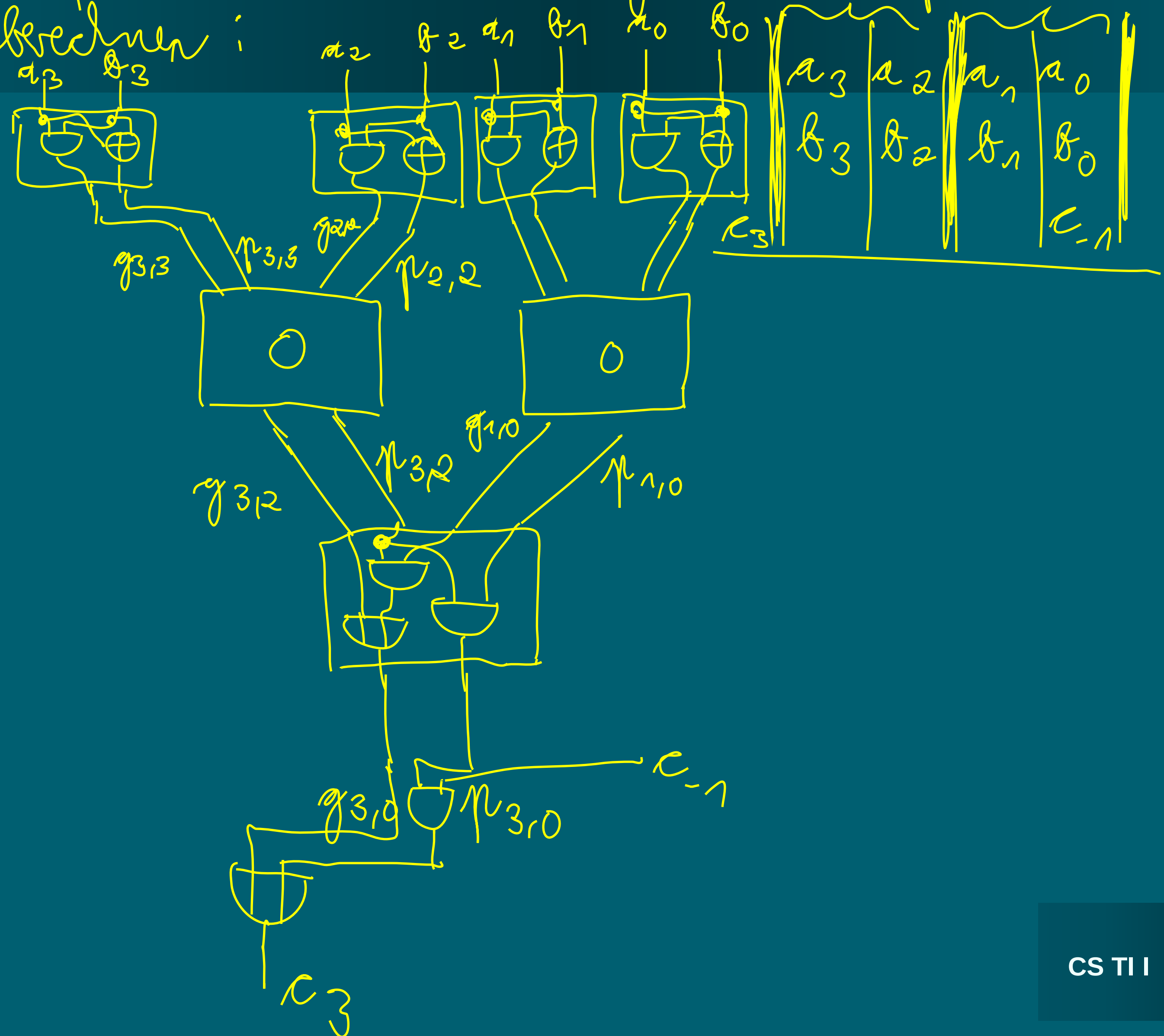
   $p_{j,i} = p_{k,i} \wedge p_{j,k+1}$

3) Für $0 \leq i < n$:

   $c_i = g_{i,0} + p_{i,0} \cdot c_{-1}$



Bsp. für $i = 3$, d.h. $g_{3,0}$ und $p_{3,0}$ sind zu berechnen:



1. Möglichkeit:

Berechne alle $g_{i,0}$, $p_{i,0}$ getrennt durch rekursiven Aufbau (d. h. Teilen in der Mitte)

$\Rightarrow$ $(g_{i,0}, p_{i,0})$ lässt sich berechnen mit Kosten $O(i)$ und Tiefe $O(\log i)$

$\Rightarrow$ insgesamt $(\forall i)$ Kosten $O(n^2)$, Tiefe $O(\log n)$

Eigenschaften von $g_{j,i}$, $p_{j,i}$ (ff)

Es genügt also, $g_{i,0}$, $p_{i,0}$ $(0 \leq i < n)$ zu berechnen.

$$s_i = a_i \oplus b_i \oplus c_{i-1} = p_{i,i} \oplus \left(g_{i-1,0} + p_{i-1,0} \cdot c_{-1}\right) \quad \text{für } 1 \leq i < n$$

$$s_n = c_{n-1} = g_{n-1,0} + p_{n-1,0} \cdot c_{-1}$$

$$s_0 = a_0 \oplus b_0 \oplus c_{-1} = p_{0,0} \oplus c_{-1}$$

Um parallele Präfixberechnung anwenden zu können, benötigt man einen geeigneten assoziativen Operator $\circ$

Der Operator $\circ$

$\circ : M \times M \to M$

Wähle

$M = (\mathbf{B}_{2n})^2, \qquad \circ : (\mathbf{B}_{2n})^2 \times (\mathbf{B}_{2n})^2 \to (\mathbf{B}_{2n})^2$ mit

$$(g_2, p_2) \circ (g_1, p_1) = (g_2 \vee (g_1 \wedge p_2), p_1 \wedge p_2)$$

$$(g_{j,i}, p_{j,i}) = (g_{j,k+1} \vee (g_{k,i} \cdot p_{j,k+1}), p_{j,k+1} \cdot p_{k,i})$$

$$= (g_{j,k+1}, p_{j,k+1}) \circ (g_{k,i}, p_{k,i})$$

Der Operator $\circ$

Wähle

$$M = (\mathbf{B}_{2n})^2, \qquad \circ : (\mathbf{B}_{2n})^2 \times (\mathbf{B}_{2n})^2 \to (\mathbf{B}_{2n})^2 \qquad \text{mit}$$

$$(g_2, p_2) \circ (g_1, p_1) = (g_2 \vee (g_1 \wedge p_2), p_1 \wedge p_2)$$

Damit läßt sich 2) schreiben als

$$(g_{j,i}, p_{j,i}) = (g_{j,k+1}, p_{j,k+1}) \circ (g_{k,i}, p_{k,i})$$

Der Operator $\circ$

Wähle

$$M = (\mathbf{B}_{2n})^2, \qquad \circ : (\mathbf{B}_{2n})^2 \times (\mathbf{B}_{2n})^2 \to (\mathbf{B}_{2n})^2 \qquad \text{mit}$$

$$(g_2, p_2) \circ (g_1, p_1) = (g_2 \vee (g_1 \wedge p_2), p_1 \wedge p_2)$$

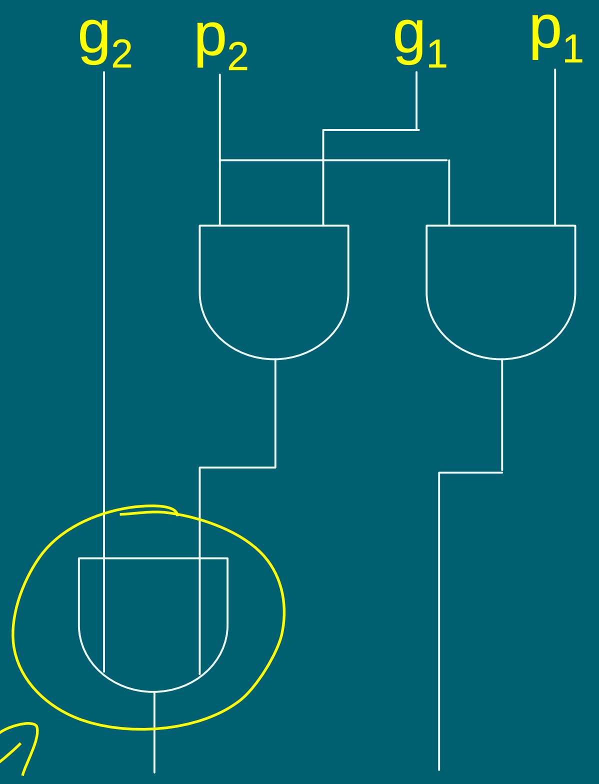
Damit läßt sich 2) schreiben als

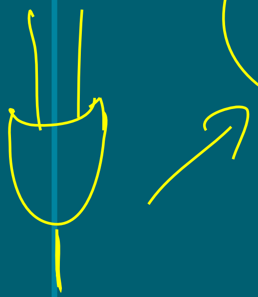
$$(g_{j,i}, p_{j,i}) = (g_{j,k+1}, p_{j,k+1}) \circ (g_{k,i}, p_{k,i})$$

Es gilt: $C(\circ) = 3$

$\text{depth}(\circ) = 2$

Basiszelle der Operation ○





Lemma 10.3

Die Operation $\circ$ ist assoziativ.

z.z.: $\forall\ (g_3, p_3), (g_2, p_2), (g_1, p_1) \in (B_{2n})^2$

<u>Beweis</u>: gilt $((g_3, p_3) \circ (g_2, p_2)) \circ (g_1, p_1) = (g_3, p_3) \circ ((g_2, p_2) \circ (g_1, p_1))$

Nachrechnen unter Verwendung von Gesetzen der Booleschen Algebra.

Bedeutung für die parallele Präfixberechnung

$(g_{i,0}, p_{i,0}) = [(g_{i,i/2}, p_{i,i/2})] \circ [(g_{i/2-1,0}, p_{i/2-1,0})] =$

Aus 2) und der Assoziativität folgt:

$(g_{i,i}, p_{i,i}) \circ (g_{i-1,i-1}, p_{i-1,i-1}) \circ \dots \circ (g_{0,0}, p_{0,0})$

$$\left(g_{i,0}, p_{i,0}\right) = \left(g_{i,i}, p_{i,i}\right) \circ \dots \circ \left(g_{1,1}, p_{1,1}\right) \circ \left(g_{0,0}, p_{0,0}\right)$$

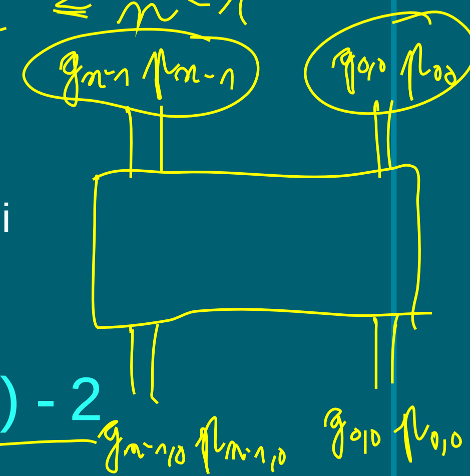
P. Präfix: $y_i = x_i \circ \dots \circ x_0$ für $0 \le i \le n-1$

Wende nun parallele Präfixberechnung an.

$\left(g_{i,0}, p_{i,0}\right)$ $(0 \le i < n)$ lassen sich aus $g_{i,i}$, $p_{i,i}$

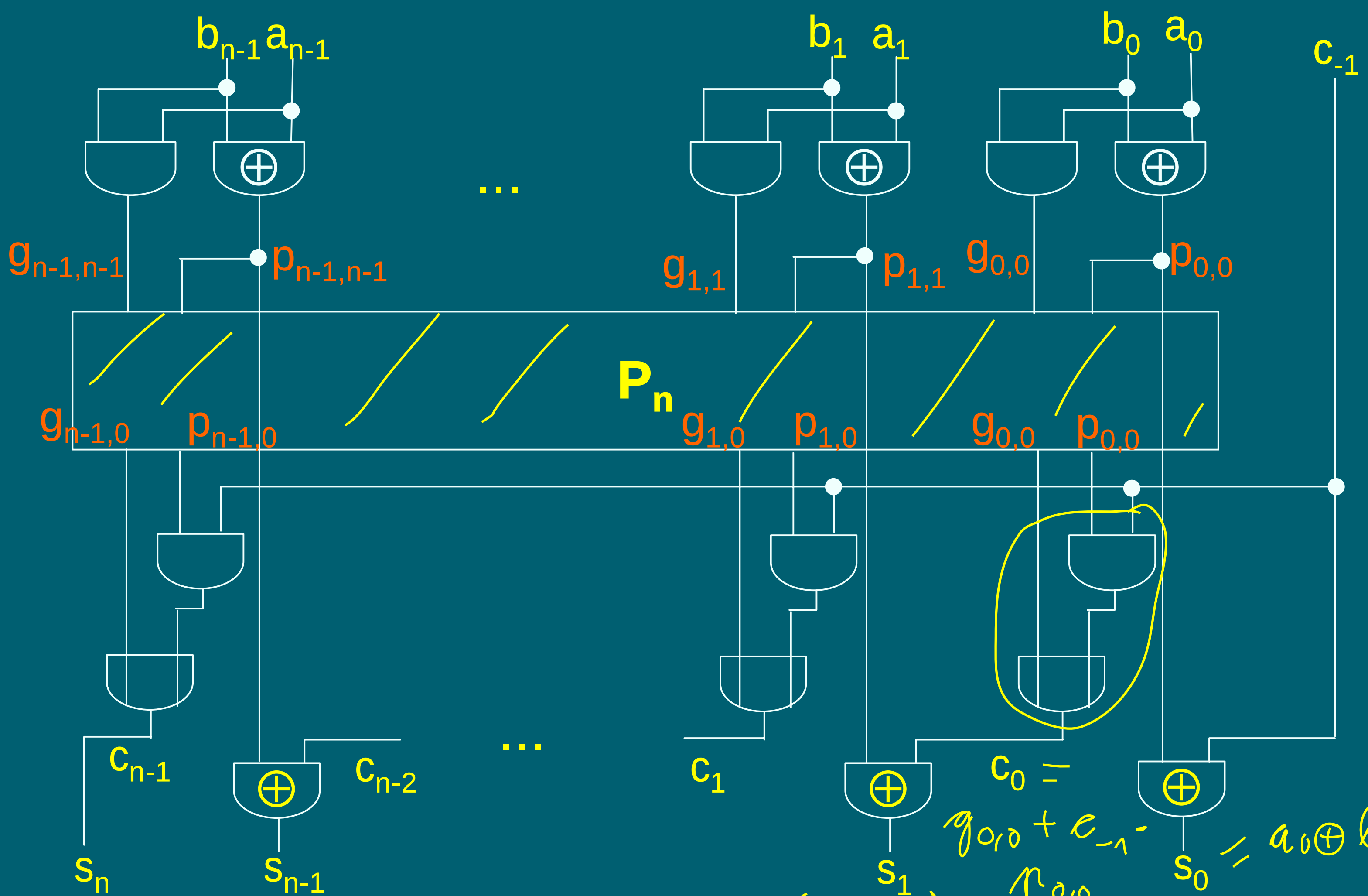
bestimmen mit Kosten $\le 2n \cdot C(\circ) = 6n$

und Tiefe $(2\log(n) - 1) \cdot \mathrm{depth}(\circ) = 4\log(n) - 2$



Gesamtschaltkreis

$$\begin{array}{lll} a_{n-1} \;\ldots & a_1 & a_0 \\ b_{n-1} & b_1 & b_0 \\ & c_0 & c_{-1} \\ \hline s_n \;\; s_{n-1} & & s_0 \end{array}$$



$c_0 = g_{0,0} + c_{-1} \cdot p_{0,0}$

$s_0 = a_0 \oplus b_0 \oplus c_{-1}$

$s_1 = (a_1 \oplus b_1) \oplus c_0 = p_{1,1} \oplus c_0$

Gesamtkosten

Kosten: $C(CLA_n) \leq \underline{6n} + 2n + 3n = 11n$

Tiefe: $\underline{depth(CLA_n)} \leq \underline{4 \log(n) - 2} \; \underline{+1} \; \underline{+3}$

$= \underline{4 \log(n) + 2}$