

- a. Die Korrekturfaktoren für positive ganzzahlige Shiftfolgen sind zu berechnen.
- b. In Einzelschritten ist der Rechenablauf zur Bestimmung von $\cos z_0$ und $\sin z_0$ mit $z_0 = .10110$ anzugeben.
- c. Ein CORDIC mit der vorgegebenen Genauigkeit zur Korrektur des Skalierungsfaktors von positiven ganzzahligen Shiftfolgen ist zu ermitteln.
- d. Es ist der Ablauf zur Bestimmung von e^{z_0} und \sqrt{w} zu beschreiben.

4 Maßnahmen zur Leistungssteigerung

Zur Erzielung zuverlässiger und voll funktionsfähiger Schaltungen werden digitale Systeme synchron getaktet. Insbesondere das sichere Zusammenspiel zwischen Datenpfad und Steuerung wird hierdurch in komplexen Schaltungen ermöglicht. Jede Datenübernahme der Ergebnisse von Teilschaltungen in speichernde Register geschieht zu einem fest definierten Zeitpunkt. Alle Operanden eines Funktionsblocks stehen zeitgleich zur Verfügung. Bei wortorientierter Verarbeitung gilt dies auch für die einzelnen Bits. Die Vorteile der synchronen Arbeitsweise liegen bei der Zuverlässigkeit der Funktionsweise und bei der Vereinfachung des Entwurfs. Zwei Nachteile seien genannt. Es muß dafür Sorge getragen werden, daß auch in großen Systemen allen Teilschaltungen der Takt zeitgleich zur Verfügung gestellt wird. Durch besondere Maßnahmen müssen die Wirkungen von zeitlichem Taktversatz (Takt-Skew) ausgeglichen werden. Ein weiterer Nachteil ist, daß die Periodendauer des Taktes an das Verzögerungsverhalten der langsamsten Teilschaltung angepaßt werden muß. Eine Laufzeitkompensation zwischen schnellen und langsamen Teilschaltungen ist nicht möglich. Die langsamste Schaltung bestimmt die Datendurchsatzrate.

In dem nachfolgenden Abschnitt werden Architekturmaßnahmen zur Erhöhung der Durchsatzrate vorgestellt. Einen breiten Raum nimmt dabei das Pipelining ein. In einem zweiten Abschnitt wird ein Maß für den Effizienzvergleich verschiedener Schaltungen erläutert.

4.1 Parallelverarbeitung und Pipelining

Für die Bestimmung der Durchsatzrate eines Systems ist die Verzögerungszeit der langsamsten Teilschaltung maßgebend. Auch das speichernde Register liefert einen Beitrag zur Verzögerungszeit. Zur Erläuterung hierzu wird Bild 4.1.1 verwendet. Es zeigt einen Ausschnitt aus einer größeren synchronen Schaltung. Eine kombinatorische Logik F_i ist mit zwei dynamischen D-FFs zusammengeschaltet. Das Eingang-D-FF liefert fortlaufend, taktsynchron Daten an die Teilschaltung. Die Ergebnisse der Teilschaltung werden wie die Eingangsdaten fortlaufend, taktsynchron übernommen und weitergeleitet. Ein dynamisches D-FF ist eine Vereinfachung des quasistatischen D-FFs von Bild 2.2.13. Diese Vereinfachung bietet sich in MOS-Schaltungen mit kontinuierlicher Taktung bei Taktfrequenzen im MHz-Bereich an. Es wird ein nichtüberlappender Zweiphasentakt angenommen. Eine derartige Tak-

tung ist besonders sicher und wird häufig innerhalb integrierter Schaltungen verwendet [16], [17], [18], [19]. Die Periodendauer des Taktes muß folgende Bedingung erfüllen:

$$T_{CLK} \geq T_{D,\phi_2} + T_{D,F_i} + T_{D,\phi_1} + T_{\phi_1,\phi_2} \quad (4.1.1)$$

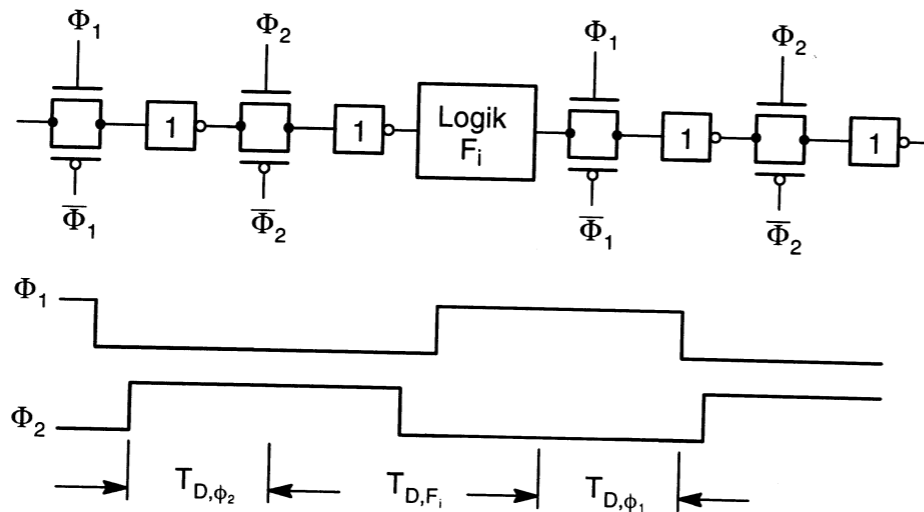


Bild 4.1.1: Ansteuerung und Datenübernahme bei einem Logikblock F_i unter Verwendung dynamischer D-FFs. Blockschaltbild und Zeitdiagramm

Hierbei ist T_{D,ϕ_2} die Verzögerungszeit des Datums zwischen dem durch Φ_2 getakteten Transmission-Gate und dem Eingang der Teilschaltung F_i . T_{D,F_i} ist die Verzögerungszeit der Teilschaltung F_i . T_{D,ϕ_1} ist die Verzögerungszeit zwischen Ausgang F_i und dem Invertereingang hinter dem durch Φ_1 getakteten Transmission-Gate. T_{ϕ_1,ϕ_2} ist die Taktpause zwischen Φ_1 und Φ_2 . Es ist zu beachten, daß die Verzögerungszeiten T_{D,ϕ_2} und T_{D,ϕ_1} von Kenndaten der Teilschaltung abhängen. Die Eingangskapazität von F_i beeinflusst T_{D,ϕ_2} und der Ausgangswiderstand von F_i beeinflusst T_{D,ϕ_1} .

Die durch D-FFs verursachte Gesamtverzögerung ist gegeben durch

$$T_{D,FF} = T_{D,\phi_2} + T_{D,\phi_1} + T_{\phi_1,\phi_2} \quad (4.1.2)$$

Eine entsprechende Verzögerung kann auch für andere Taktungssysteme angegeben werden. In Einphasentaktssystemen mit taktflankengetriggerten FFs ist beispielsweise die Summe von Hold- und Set-Up-Zeit einzusetzen.

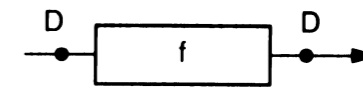


Bild 4.1.2: Vereinfachte Symbolik für die D-FFs (Delayoperator D) am Eingang und Ausgang einer Funktionseinheit f

Nachfolgend wird eine vereinfachte Symbolik für synchron getaktete Funktionseinheiten nach Bild 4.1.2 verwendet. Hierbei werden die D-FFs nur durch einen Punkt in der Leitung symbolisch dargestellt.

Die erzielbare Durchsatzrate R_T eines Systems in Bit pro Zeiteinheit ist proportional zur Taktfrequenz, d.h.

$$R_T \sim \frac{1}{T_{CLK}} \quad (4.1.3)$$

Die die maximale Durchsatzrate bestimmende Periodendauer wiederum wird durch die ungünstigste Teilschaltung bestimmt.

$$T_{CLK} = \max_i (T_{D,F_i} + T_{D,FF_i}) \quad (4.1.4)$$

Für eine hohe Durchsatzrate sind kleine Verzögerungen notwendig. Geringe Verzögerungen können durch technologische Maßnahmen erreicht werden. Durch Verkleinerung der geometrischen Strukturen (Skalierung) können die Kapazitäten reduziert und hierdurch eine Verringerung der Verzögerungszeit erzielt werden. In der Literatur werden die Effekte einer solchen Skalierung diskutiert [17], [18].

Neben technologischen Maßnahmen sind auch schaltungstechnische Maßnahmen zur Erhöhung der Durchsatzrate möglich. Im Kapitel 3 wurden verschiedene Schaltungsstrukturen für die Realisierung von Basisoperationen dargestellt. Die gezeigten Alternativen zeigen ein unterschiedliches Verhalten der Verzögerungszeit. Nach (4.1.4) ist das Maximum der Verzögerungszeit von Teilschaltungen zu minimieren. Dies bedeutet, daß nur das langsamste Modul verbessert werden muß. Beispielweise müßten somit bei einer Signalverarbeitung mit mehrfacher Addition und Multiplikation nur die Multiplizierer lauffeitmäßig optimiert werden. Für die Addierer würde dies keinen Sinn machen. Gesucht sind nun Architekturmaßnahmen zur Erhöhung der Durchsatzrate, bei denen die Dominanz des langsamsten Moduls durchbrochen werden kann.

Naheliegender scheint, daß, wenn ein Modul eine geforderte Durchsatzrate nicht bereitstellen kann, gleichzeitig mehrere Module die Bearbeitung durchführen müssen. Eine parallele Verarbeitung nach Bild 4.1.3 löst dieses Problem. Der Block DMUX verteilt die zu bearbeitenden Daten auf parallele gleiche Module. Der Block MUX faßt die parallelen Ergebnisse wieder zu einem Datenstrom zusammen. Die Daten werden in den parallelen Modulen entweder mit einem zeitlichen Versatz von

T_{CLK} / N bearbeitet oder die Einheiten DMUX und MUX müssen so mit Verzögerungseinheiten für Daten versehen werden, daß eine zeitgleiche Bearbeitung möglich ist. Die gezeigte Struktur hat ferner die Voraussetzung, daß die Operationen der Einheit f nicht vom Ergebnis früherer Operationen abhängen. Die Durchsatzrate eines Systems mit N Pfaden ist das N -fache des einfachen Systems.

$$R_{T,N} = N \cdot R_{T,1} \quad (4.1.5)$$

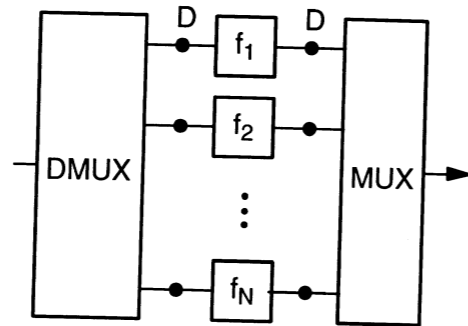


Bild 4.1.3: Parallele Implementierung einer Funktion f durch N gleichartige Funktionseinheiten

Die parallele Verarbeitung ist nicht nur für N gleichartige, sondern auch für unterschiedliche Teilmodule möglich. Im Einzelfall muß immer die Datenabhängigkeit der Teilfunktion und die Bildung des Ergebnisses betrachtet werden. Als ein einfaches Beispiel sei

$$y = F(x) = \sum_{i=1}^N f_i(x) \quad (4.1.6)$$

betrachtet. Die Datenabhängigkeit ist sehr einfach. Alle Teilfunktionen erhalten das gleiche Argument x , d.h. die Eingangsdaten x werden parallel allen Teilmodulen zugeführt. Die Teilmodule werden parallel realisiert und das Gesamtergebnis wird durch Addition der Teilergebnisse gebildet. Die Teilmodule arbeiten zwar parallel, jedoch ist ein abschließender Multioperandenaddierer wesentlich für die Durchsatzrate. Eine derartige Anordnung ist in Bild 4.1.4 gezeigt. Die Durchsatzrate wird von dem langsamsten Teilmodul f_i und vom Multioperandenaddierer bestimmt.

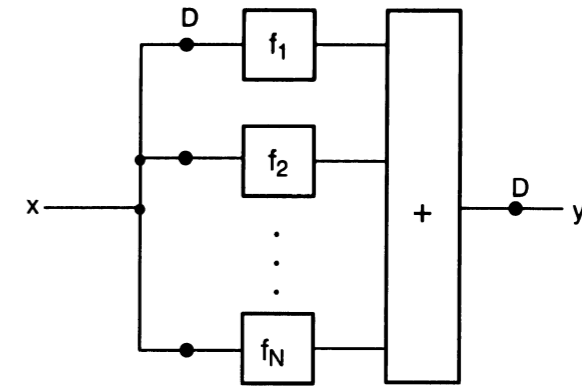


Bild 4.1.4: Parallele Realisierung der Funktion $y = \sum_{i=1}^N f_i(x)$

$$R_{T,N} \sim \frac{1}{\max_i T_{D,f_i} + T_{D,ADD} + T_{D,FF}} \quad (4.1.7)$$

Als Alternative zu der Parallelisierung ist es in vielen Fällen möglich, eine Funktion F in eine serielle Zusammenschaltung von Teilfunktionen f_i zu entwickeln. Das Ergebnis einer Teilfunktion wird als Eingangssignal der folgenden Teilfunktion verwendet.

$$F = f_N(f_{N-1} \dots f_2(f_1(\cdot)) \dots) \quad (4.1.8)$$

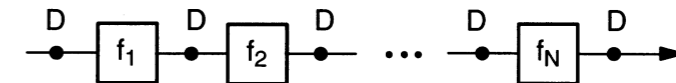


Bild 4.1.5: Realisierung einer Funktion durch eine im Pipelining betriebene Kaskade von Teilfunktionen f_i

Durch Einfügung von synchron getakteten Zwischenspeichern entsteht eine Architektur nach dem Fließbandprinzip (Bild 4.1.5). Entsprechend dem englischen Sprachgebrauch wird auch hier die Fließbandverarbeitung als Pipelining bezeichnet. Während ein Teilmodul das Teilergebnis der Vorgängerstufe übernimmt und bearbeitet, erhält die Vorgängerstufe zeitgleich ein neues Datum zur Bearbeitung. Das langsamste Teilmodul der Kette bestimmt die Durchsatzrate. Für die Durchsatzrate gilt:

$$R_{T,N} \sim \frac{1}{\max_i T_{D,f_i} + T_{D,FF}} \quad (4.1.9)$$

$$R_{T,N} = R_{T,1} \cdot N \frac{T_{D,J} + T_{D,FF}}{T_{D,J} + N T_{D,FF}} \quad (4.1.10)$$

Dies bedeutet, daß insbesondere für große N aufgrund der Verzögerungszeit der Register die Durchsatzrate nicht mehr proportional zu N zunimmt. Methoden zur Gewinnung von Pipeline-Architekturen sollen nun weiter untersucht werden. Das zeitliche Verhalten synchron getakteter Logikblöcke soll für die folgenden Betrachtungen weiter abstrahiert werden. Bei korrekter Wahl der Taktrate wird die Verzögerungszeit der Logikfunktionsblöcke immer kleiner sein als die Periodendauer des Taktes abzüglich von Übergangszeiten der Flip-Flops (4.1.1). Nach außen hin wirkt diese Anordnung so, als ob ein Eingangssignal des Funktionsblockes mit einer Verzögerung von einer Taktperiode zu dem Ergebnis am Ausgang des Datenübernahme-Flip-Flops führt. Ein gleiches Verhalten wird beschrieben, wenn man den Logikblock als verzögerungsfrei betrachtet und die Gesamtverzögerung einer Taktperiode nur dem Datenübernahme-Flip-Flop zuordnet. Die Funktion eines D-FFs kann folglich abstrakt als ein Delay-Operator behandelt werden.

Es sei $D[\cdot]$ ein Delay-Operator mit der Verzögerung einer Taktperiode T .

$$D[f(t)] = f(t - T) \quad (4.1.11)$$

Eine mehrfache Anwendung des Delay-Operators sei beschrieben durch

$$D^n[f(t)] = f(t - nT) \quad (4.1.12)$$

Es sei darauf hingewiesen, daß der Delay-Operator hier im Zeitbereich definiert ist. In der Literatur zur Signalverarbeitung werden derartige Delay-Operatoren meist über die Z -Transformierte beschrieben.

$$\begin{aligned} f(t) &\bullet\text{---}\circ F(z) \\ f(t - n) &\bullet\text{---}\circ z^{-n} F(z) \end{aligned} \quad (4.1.13)$$

Es wird dabei die Taktperiode auf 1 normiert und die Delay-Operation entspricht dann dem Produkt mit z^{-1} . Beide Beschreibungsformen der Delay-Operation in getakteten Systemen werden hier nebeneinander verwendet.

Mit dem definierten Delay-Operator gilt für einen Funktionsblock mit 2 Eingangsvariablen x_1, x_2 der Zusammenhang

$$\begin{aligned} f(t) &= f(x_1(t), x_2(t)) \\ y(t) &= D[f(t)] = f(t - T) \end{aligned} \quad (4.1.14)$$

wobei y das Ergebnissignal am Ausgang des Übernahme-FFs ist (Bild 4.1.6). Es ist offensichtlich, daß die Distributivität von Funktionsoperation und Delay-Operation gilt, d.h.

$$\begin{aligned} D[f(x_1(t), x_2(t))] &= f(D[x_1(t)], D[x_2(t)]) \\ &= f(x_1(t - T), x_2(t - T)) \end{aligned} \quad (4.1.15)$$

Dies bedeutet, daß in einem getakteten System die Verzögerung des Ergebnisses durch ein D-FF die gleiche Wirkung hat, wie die Verzögerung der Argumente. Wie in Bild 4.1.6 gezeigt kann dies als ein Delay-Transfer vom Ausgang zum Eingang und umgekehrt beschrieben werden. Der Delay-Transfer kann formal auch als Addition und Subtraktion von Verzögerungszeiten T behandelt werden [60]. Wird bei dem linken Teil von Bild 4.1.6 am Ausgang T subtrahiert und am Eingang T addiert, so erhält man den rechten Teil von Bild 4.1.6.

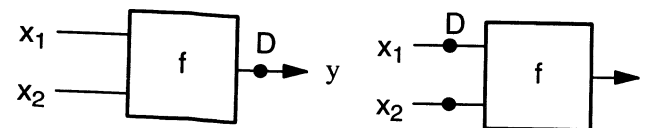


Bild 4.1.6: Delay-Transfer zwischen Ausgang und Eingang

Neben dem Delay-Transfer ist die Zeitskalierung eine wichtige Maßnahme bei der Gewinnung von Pipeline-Architekturen [61]. Durch Überabtastung mit einem ganzzahligen Faktor n kann ein Verzögerungsintervall T in mehrfache Verzögerungen in einem mit dem Faktor n skalierten System umgewandelt werden.

$$T = n T' \quad (4.1.16)$$

Dies bedeutet jedoch auch, daß ein D-FF in dem ursprünglichen System in n D-FFs in einem System mit n -facher Überabtastung verwandelt werden kann. Wie in Bild 4.1.7 gezeigt, können die beiden Regeln Delay-Transfer und Skalierung kombiniert werden.

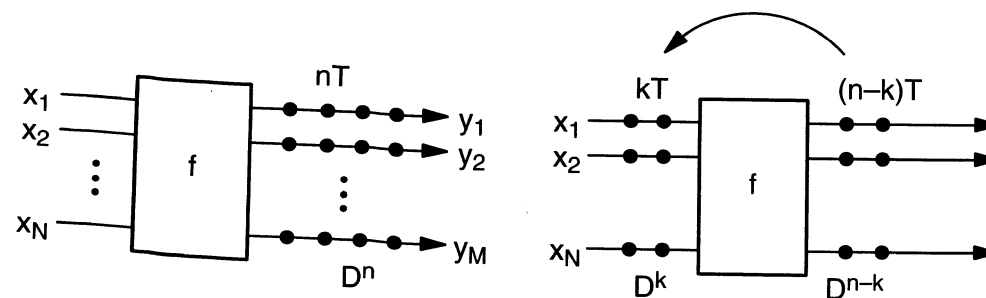


Bild 4.1.7: Verallgemeinerter Delay-Transfer. D^n n -facher Delayoperator, nT zugehörige Verzögerungszeit

Im Bild 4.1.6 und im Bild 4.1.7 wurde das Funktionsmodul durch einen einheitlichen Block charakterisiert. Dieser Block kann sich nun, wie in Bild 4.1.8 angedeutet, aus einem Netzwerk von mehreren Teilfunktionsblöcken (Prozessorelementen) zusammensetzen.

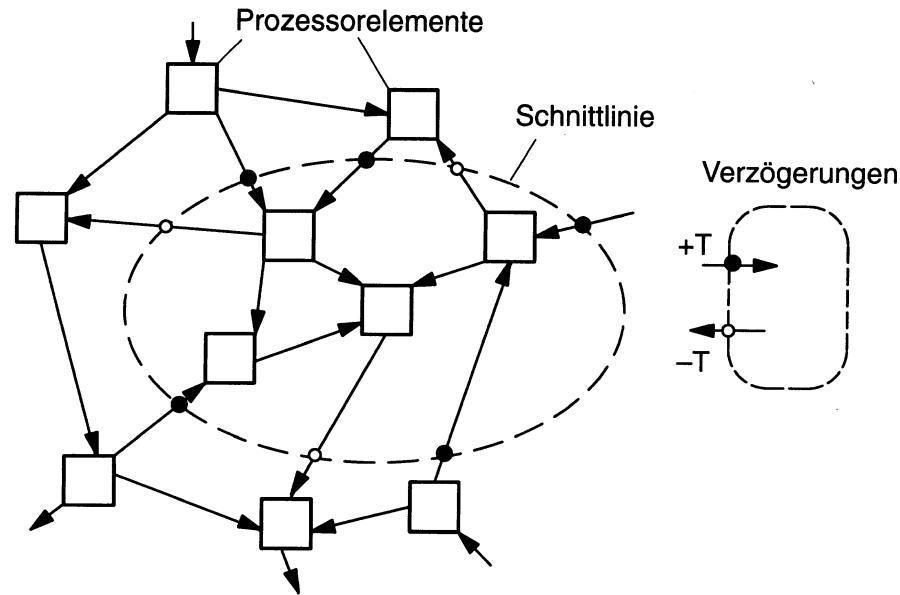


Bild 4.1.8: Darstellung der Cut-Set-Methode. Wirkungsneutrale Einfügung von Delays an der Schnittlinie einer Trennmenge von Prozessorelementen

Eine Schnittlinie (engl. cut) trennt die Menge der Prozessorelemente. Die Anwendung des Delay-Transfers kann nun so erfolgen, daß alle auf die Menge zuführenden Pfade ein zusätzliches positives Delay (+ T) und alle wegführenden Pfade ein zusätzliches negatives Delay (- T) erhalten. Diese Methode wird im Englischen als "cut-set procedure" bezeichnet. Die Cut-Set-Methode führt dazu, daß trotz der Delay-Transfers nach außen hin das System das gleiche Verhalten zeigt [61]. Negative Verzögerungen gibt es nicht. Daher müssen im Regelfall die negativen Verzögerungen durch in Prozessorelementen vorhandene Verzögerungen kompensiert werden. Am Eingang und Ausgang eines Systems können negative Verzögerungen speziell behandelt werden. Eine Verzögerung (- nT) am Eingang kann durch eine um n Takte frühere Datenzufuhr realisiert werden. Eine Verzögerung (- nT) am Ausgang kann weggelassen werden. Dies führt zu einer Verzögerung von n Takten. Letztlich führt das Weglassen der negativen Verzögerungen am Eingang bzw. Ausgang zu einer Erhöhung der Latenzzeit eines Systems. Dieser Zusammenhang ist in Bild 4.1.9 gezeigt.

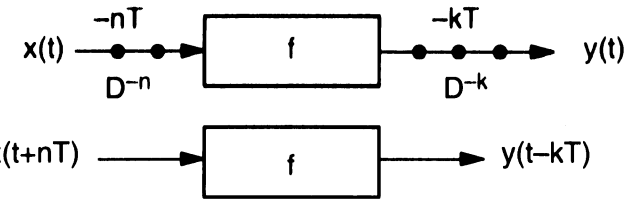


Bild 4.1.9: Elimination negativer Verzögerungen am Eingang und Ausgang von Funktionsblöcken

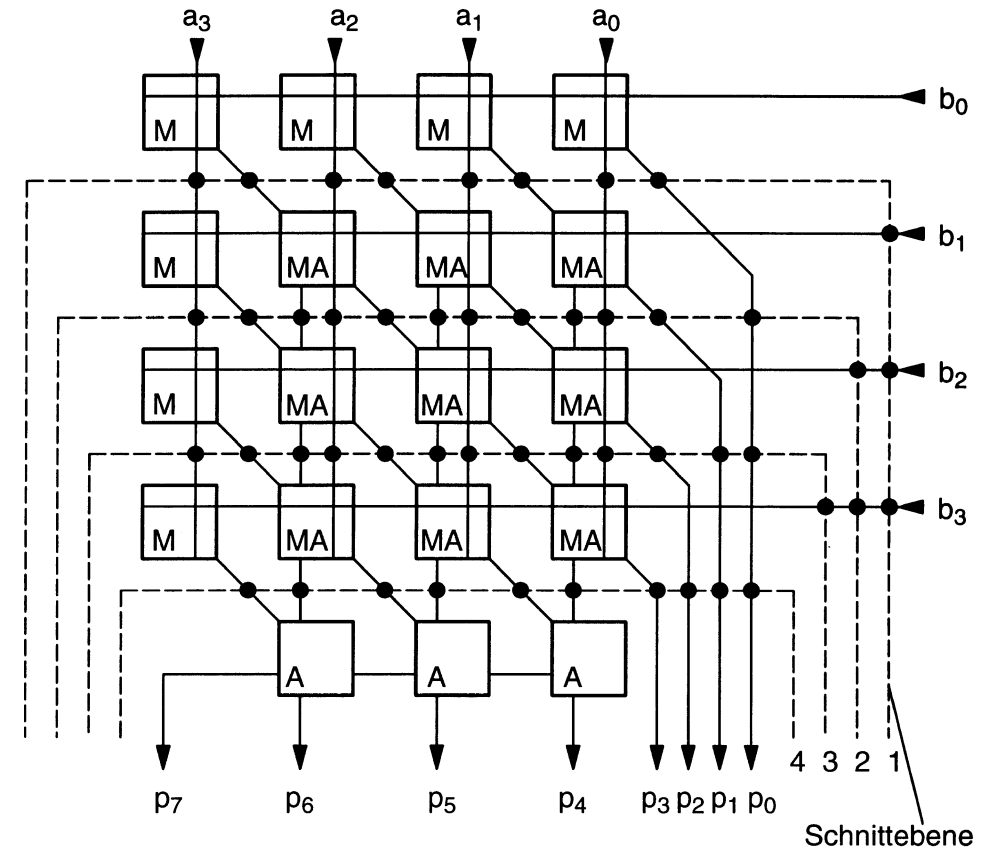


Bild 4.1.10: Bildung von Pipeline-Stufen mit der Cut-Set-Methode am Beispiel eines 4 · 4 bit Carry-Save-Array-Multiplizierers

In vielen Fällen wird bei der Cut-Set-Methode der Delay-Transfer mit der Zeitskalierung kombiniert [61]. Hierbei werden dann anstatt + T bzw. - T Bruchteile von T beispielsweise kT' mit $0 < kT' < T$ addiert bzw. subtrahiert. Das so gewonnene System muß in einem Überabtastmode betrieben werden.

Die erwähnte Cut-Set-Methode soll an zwei Beispielen angewandt werden. In Bild 4.1.10 ist die Cut-Set-Methode für das Beispiel eines 4 · 4 bit Carry-Save-Array-Multiplizierers gezeigt. Das Array ist abweichend zu Bild 3.3.5 nicht in Rautenform, sondern in Rechteckform gebildet. Die Prozessorzellen vom Typ M sind 1 · 1 bit Multiplizierer und die vom Typ A sind Addiererzellen (Voll- bzw. Halb-addierer). Die Prozessorzelle vom Typ MA ist die Kombination von beiden. Die Schnittebenen sind so gewählt, daß das Array horizontal geschnitten wird und die Schnittebenen zum Ausgang hin geschlossen werden. Alle Signalübergänge an den Schnittebenen im Array haben die gleiche Richtung. Hier werden D-FFs eingefügt. Die Produktleitungen am Ausgang sind entsprechend der Anzahl der Schnittebenen verzögert. Es ist zu beachten, daß auch die Operandenbits durch D-FFs verzögert werden.

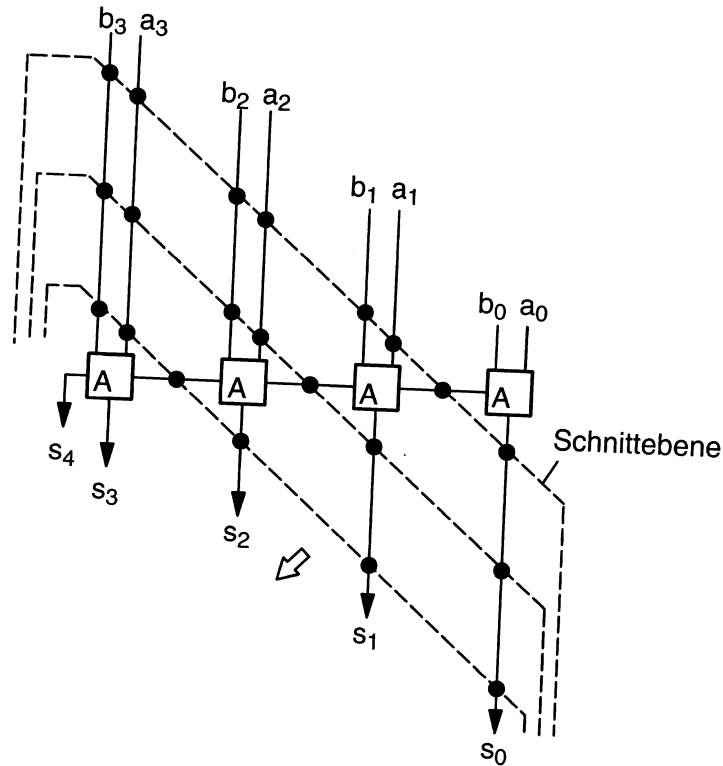


Bild 4.1.11: Bildung von Pipeline-Stufen mit der Cut-Set-Methode am Beispiel eines 4 bit Carry-Ripple-Addierers

In Bild 4.1.11 ist gezeigt wie bei einem Carry-Ripple-Addierer D-FFs zum Pipelining eingefügt werden. Der zeitkritische Pfad läuft den Übertragungspfad entlang. Deshalb müssen hier FFs eingefügt werden. Die Cut-Set-Methode zeigt jedoch, daß

auch die zugeführten Operanden und die Summenbits verzögert werden müssen. Damit Übertragsbit und Operandenbits zeitgleich bei einer Addiererzelle eintreffen, ist eine entsprechende Verzögerung der Operandenbits erforderlich (Pre-Skewing). Damit alle Summenbits zeitgleich an eine nachfolgende Einheit übergeben werden, sind frühzeitig ermittelte Summenbits zu verzögern (De-Skewing). Die Cut-Set-Methode zeichnet sich durch eine elegante automatische Ermittlung der erforderlichen Verzögerungen von Operandenbits und Ergebnisbits aus.

4.2 Effizienzvergleich

Die Optimierung von Architekturen hat die Erhöhung der Leistungsfähigkeit zum Ziel. Als Maß für die Leistungsfähigkeit wurde im vorigen Abschnitt die Durchsatzrate benannt. Die Bestimmung der Durchsatzrate ist häufig problematisch. Beispielsweise nimmt im Laufe der Signalverarbeitung die Wortbreite zu, so daß für den Vergleich von Architekturen eine spezielle Schnittstelle als Vergleichspunkt genommen werden muß, die allen gemeinsam ist. Hier bietet sich die Datenzufuhrrate oder die Ergebnisrate an. Für die Signalverarbeitung wird in vielen Fällen statt der Durchsatzrate R_T die Rechenleistung R_C in Operationen pro Zeiteinheit als Maß für Leistungsfähigkeit genommen. Es soll jedoch auch hier angemerkt werden, daß bei der Rechenleistung die zugrunde gelegte Wortbreite zu beachten ist. Es ist naheliegend, daß z.B. 8 bit und 32 bit Operationen von der Leistungsfähigkeit her nicht als gleich eingestuft werden können.

In synchron getakteten Systemen ist die Periodendauer T_{CLK} des Taktes eine Bezugsgröße zur Bestimmung sowohl von der Rechenleistung als auch von der Durchsatzrate. Für die Rechenleistung gilt

$$R_C = \frac{n_{OP}}{T_{CLK}} \quad (4.2.1)$$

mit n_{OP} als Anzahl der im Zeitintervall T_{CLK} durchgeführten Operationen. Für die Datendurchsatzrate gilt

$$R_T = \frac{n_S}{T_{CLK}} \quad (4.2.2)$$

mit n_S als Anzahl der im Zeitintervall T_{CLK} parallel zu- bzw. abgeführten Abtastwerte. Die Durchsatzrate wird dann in Abtastwerten je Sekunde angegeben. Zur Angabe in bit/s muß dieser Wert mit der Anzahl der Bits je Abtastwert multipliziert werden. Aufgrund der gemeinsamen Bezugsgröße T_{CLK} gilt vielfach die Proportionalität zwischen Rechenleistung und Durchsatzrate.

$$R_C = \frac{n_{OP}}{n_S} R_T \quad (4.2.3)$$