

ATM-Switch Prototyping mit LCAs

Bernhard Lang, Cornelius Frankenfeld, Veit Pietzuch

MAZ GmbH, Harburger Schloßstraße 6-12, 21079 Hamburg, Germany

Tel.: +49/40/76629-0, email: lang@maz-hh.de, cf@maz-hh.de, vp@maz-hh.de

Zusammenfassung

Der Beitrag behandelt die Realisierung eines ATM-Schalterbausteins, welcher als Kern leistungsfähiger Breitband-ISDN Vermittlungseinheiten Anwendung findet. Es werden die Funktionen eines ATM-Schalterbausteins und dessen Merkmale in einer ASIC-Implementierung aufgezeigt. Zum *rapid prototyping* des ASIC-Schalter wird eine LCA-Version mit eingeschränkter Funktionalität vorgestellt. Der LCA-Entwurf basiert auf dem ASIC-Design und ist mittels *selbstarbitrierendem Datenfluß* realisiert.

1 Einleitung

In der modernen ATM-Breitbandkommunikation werden neue, leistungsfähige Vermittlungseinheiten benötigt. Diese Einheiten stellen vom Anwender gewünschte Verbindungskanäle zur Verfügung. Den Kern der Vermittlungseinheiten bilden Schalterbausteine. Jeder Schalter (Switch) besitzt eine feste Anzahl von Eingängen und Ausgängen und kann ATM-Zellen von Eingängen zu beliebigen Ausgängen weiterleiten (routen). Mehrere Schalterbausteine lassen sich zu Schaltnetzen zusammenfügen. ATM-Zellen sind die elementaren Datenpakete der ATM-Breitbandnetze. Sie bestehen aus 53 Bytes, welche sich in einen 5 Byte langen Kopf (Header) und 48 Bytes Nutzdaten (Payload) aufteilen. In die Eingänge eines Schalter werden ATM-Zellen eingespeist und aufgrund einer im Kopf kodierte Zieladresse an Schalterausgänge weitergereicht (routing).

Zum Aufbau von Vermittlungseinheiten ist die frühe Verfügbarkeit von Schalterbausteinen von entscheidender Bedeutung. Erst mit funktionsfähigen Schaltern ist der Test der komplexen Vermittlungseinheiten möglich.

Der vorliegende Beitrag beschreibt den Entwurf von ATM-Schaltern mit LCAs, die zum frühen Test einer ATM-Vermittlungseinheit benötigt wurden. Der Entwurf lehnt sich an ein parallel durchgeführtes VLSI Design eines Schalters an. Der LCA Entwurf implementiert eine Teilfunktionalität des VLSI-Schalters, welche zur Bereitstellung minimaler Vermittlungsmöglichkeiten und somit zum Testen nötig waren. Im Beitrag wird zunächst

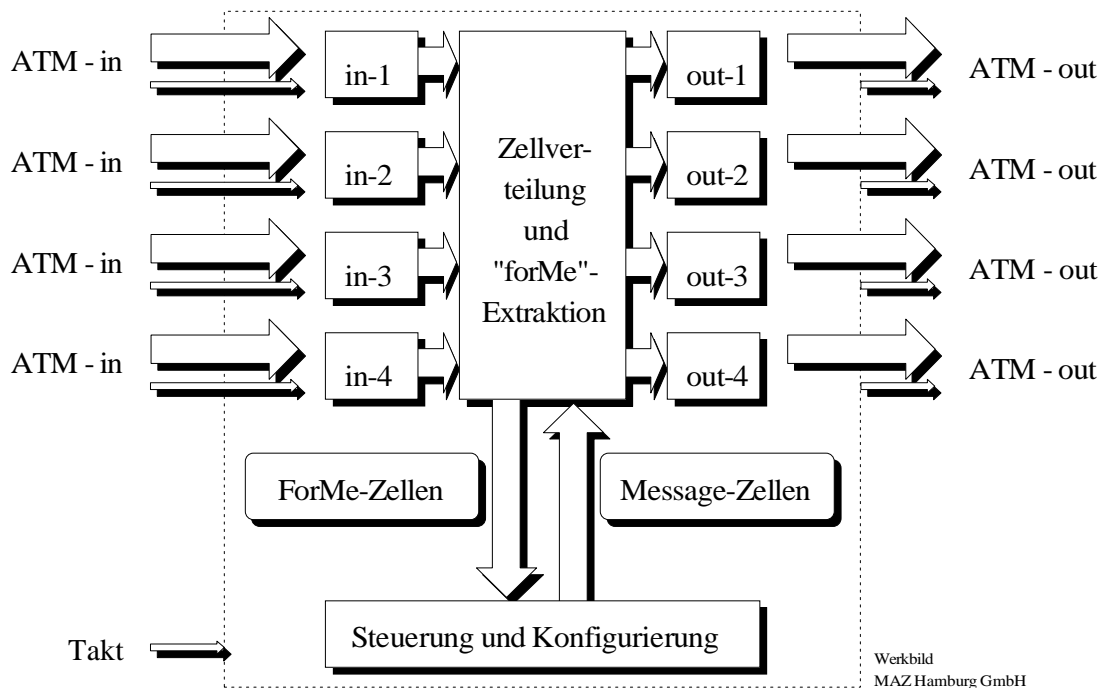


Abbildung 1: Blockschaltbild des ATM-Schalterbausteins DAS 340

das Konzept des VLSI Bausteins DAS 340 vorgestellt. Es folgt die reduzierte Architektur für das LCA Design, welches mittels *selbstarbitrierendem Datenfluß* [Lan94b] realisiert ist.

2 Der Schalterbaustein DAS 340

Mit dem DAS 340 (Abbildung 1) können 4 unabhängige Zellströme an 4 Ausgänge geroutet werden. Jeder der 4 Eingänge arbeitet mit einem separaten Takt, die interne Vermittlung der Zellen und deren Ausgabe erfolgt mit dem Haupttakt des Schalters. Die Zelldatenströme der Eingänge müssen auf den Haupttakt des Schalters synchronisiert werden.

Je nach der im Header befindlichen Routinginformation (VCI) wird eine ankommende Zelle an einen oder mehrere Ausgängen weitergeleitet. Der DAS 340 unterscheidet anhand des Headers mehrere Arten von Zellen:

Idle-Zellen werden gesendet, um einen kontinuierlichen Zellstrom zu gewährleisten und tragen keine Informationen.

User-Zellen tragen Nutzinformationen von einem Endgerät zu einem anderen Endgerät. Diese Zellen beinhalten also z.B. Sprachinformationen, Daten oder Videosignale.

ForMe-Zellen tragen Steuerinformationen innerhalb des ATM-Switchs. Diese werden i.A. vom Steuer-Rechner gesendet oder sind vom SE gesendete Antwortzellen an den

Steuer-Rechner. Durch diese Zellen werden Routinginformationen, Fehlerdiagnosen, Fehlermeldungen, Quittierungen und Registersteuerungen u.a. gesendet.

Der DAS 340 besitzt keine Steuerleitungen wie Tags oder Rechnerinterfaces. Die Steuerung geschieht ausschließlich über *ForMe*-Zellen. Um einen Mißbrauch durch nicht berechnigte Teilnehmer zu unterbinden, wurde ein Sicherheitskonzept erarbeitet. Nur bestimmte, freigegebene Eingänge des DAS 340 können *ForMe*Zellen empfangen. Nach außen wird der Switch wie durch einen Zaun abgegrenzt.

Zur Zuordnung von Zelladressen zu Ausgängen besitzt der DAS 340-Baustein eine Tabelle mit bis zu 1024 unterschiedlichen Einträgen. Diese Routing-Tabelle wird durch *ForMe*-Zellen konfiguriert und kann im laufenden Betrieb modifiziert werden.

Der DAS 340 Baustein ist als Standardzellendesign in 0.7μ Technologie realisiert und besitzt eine Komplexität von ca. 367 000 Transistoren für die Logikfunktionen. Hinzu kommen 30 kBit statischer Speicher für Pufferspeicher und Routingtabellen. Der DAS 340 arbeitet mit einem internen Systemtakt von 20 MHz.

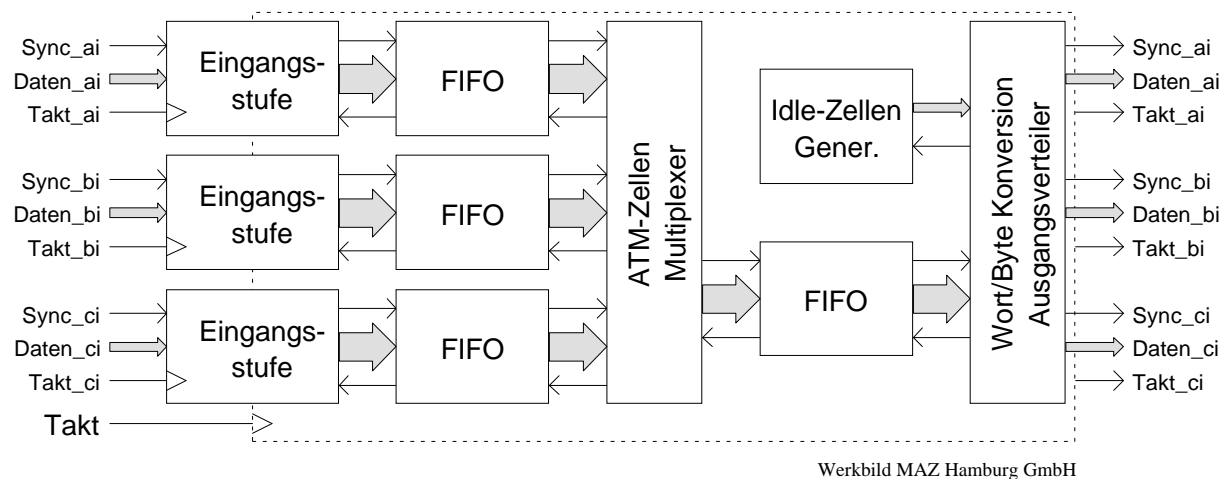


Abbildung 2: Interne Struktur des LCA-basierten ATM-Schalters SE-XIL

3 Die LCA-Version SE-XIL des ATM-Schalters

Das SE-XIL-Element ist eine vereinfachte Version des DAS 340 Schalters. Es ist in einem LCA-Baustein XC4013 implementiert. Im Vergleich zum DAS 340 sind folgende Vereinfachungen durchgeführt:

- Der Baustein ist statisch programmiert, eine Steuerung über *ForMe*-Zellen ist somit nicht möglich.
- Die Routing-Tabelle ist fest programmiert.

- Die Puffergröße ist auf ein Minimum reduziert. Es existiert ein gemeinsamer Puffer für alle Ausgänge.
- Die Anzahl der Kanäle ist auf drei beschränkt.

Die Entkopplung der Takte von Eingängen und interner Verarbeitung ist hingegen auch im SE-XIL realisiert. Somit werden drei Synchronisationseinheiten zwischen Eingängen und der Vermittlungseinheit benötigt. Abbildung 2 zeigt eine Übersicht der internen Verschaltung.

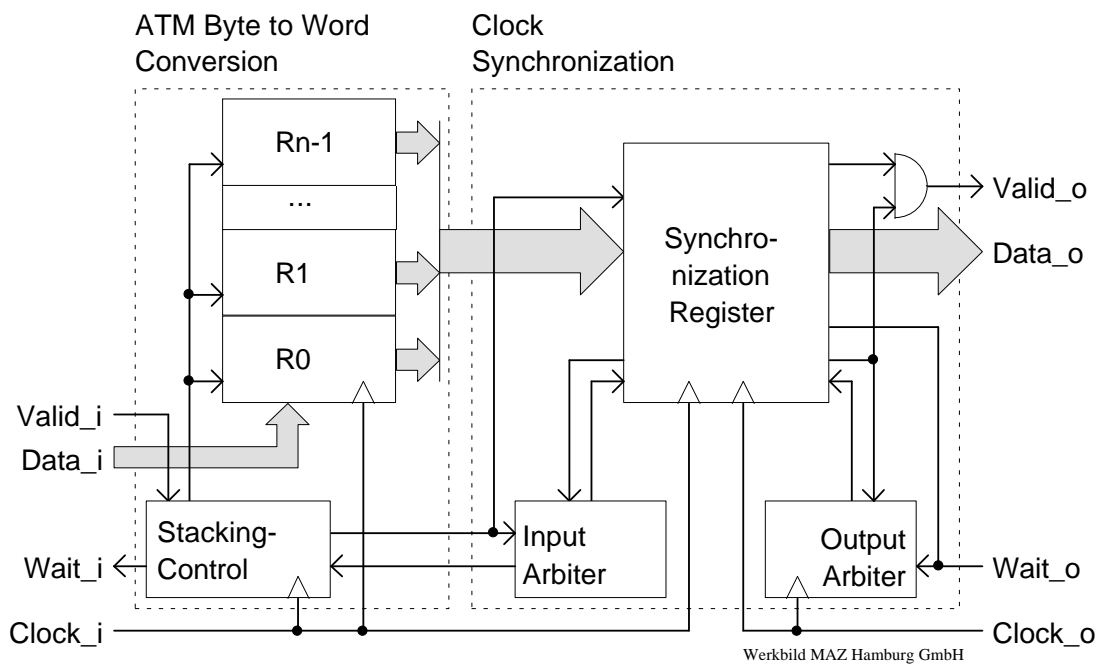


Abbildung 3: ATM-Eingang des SE-XIL Elements: Byte/Wort Konversion und Taktsynchronisation

Der Entwurf der Einzelblöcke im SE-XIL-Element basiert auf dem *selbstarbitrierenden Datenfluß*-Konzept [Lan94b]. Dieses Konzept ist dem asynchronen *Micropipeline*-Ansatz [Sut89] ähnlich, es wird ebenfalls ein *elastischer* Datenfluß modelliert, der synchrone Systemtakt jedoch beibehalten. Beim selbstarbitrierenden Datenfluß werden neben den Nutzdaten noch Signale zur Steuerung des Datenflusses mitgeführt, welche die lokale Synchronisation ermöglichen. Damit ist ein entkoppelter, elastischer Entwurf und Test einzelner Teilblöcke möglich, deren direkte Zusammenschaltung zur Gesamtfunktion führt.

Der elastische Datenfluß basiert auf gleichartigen Interfaces mit jeweils drei Signalen zwischen Teilblöcken: Über einen *Daten*-Signalvektor reicht ein Block zu verarbeitende Daten an einen nachfolgenden Block weiter. Ein *Valid*-Signal qualifiziert den zugehörigen Datenvektor als gültig. Über ein *Wait*-Signal markiert ein nachfolgender Block entgegen der Datenrichtung, ob innerhalb des aktuellen Taktzyklus die anliegenden Daten verarbeitet werden oder nicht. Die Reaktion auf gültige Daten durch deren Verarbeitung oder durch *Wait*-Signalisierung erfolgt beim *selbstarbitrierenden Datenfluß* immer innerhalb eines

Taktzyklus. Durch die offensive Weitergabe verfügbarer Daten mittels *Daten*-Signalvektor und *Valid*-Signal und durch die lokale, an die Verarbeitung angepaßte Reaktion auf diese Daten mittels *Wait*-Signal wird eine lokale Steuerung der Datenrate ermöglicht. Diese lokale Steuerung bewirkt einen elastischen, an die Verarbeitung angepaßten Datenfluß.

Zur internen Verarbeitung der ATM-Zelldaten im SE-XIL-Element werden ankommende Bytes zu 3-Byte breiten Worten zusammengefaßt; damit braucht ein Wort erst nach maximal 3 Taktzyklen verarbeitet werden. Im Anschluß an die Byte/Wort Konversion findet die Taktsynchronisation statt. Das Element zur Synchronisation ist gegenüber [Lan94b] vereinfacht, da durch die vorgeschaltete Byte/Wort Konversion nur in jedem 3. Zyklus des Eingangstakts ein gültiges Wort anliegt. Abbildung 3 zeigt das Blockschaltbild der beschriebenen Eingangsstufe.

Hinter der Synchronisation werden in jedem Eingang ankommende Worte in FIFO-Speichern (First In First Out) gesammelt, bis jeweils komplette Zellen verfügbar sind. Vollständige Zellen aller drei Ausgänge werden von einem nachfolgenden Funktionsblock auf einen gemeinsamen Ausgangsdatenstrom gemultiplext und in das Ausgangs-FIFO transferiert. Die FIFOs auf der Eingangs- und Ausgangsseite sind als Datenflußelemente realisiert und passen sich selbständig und elastisch den Datenströmen an.

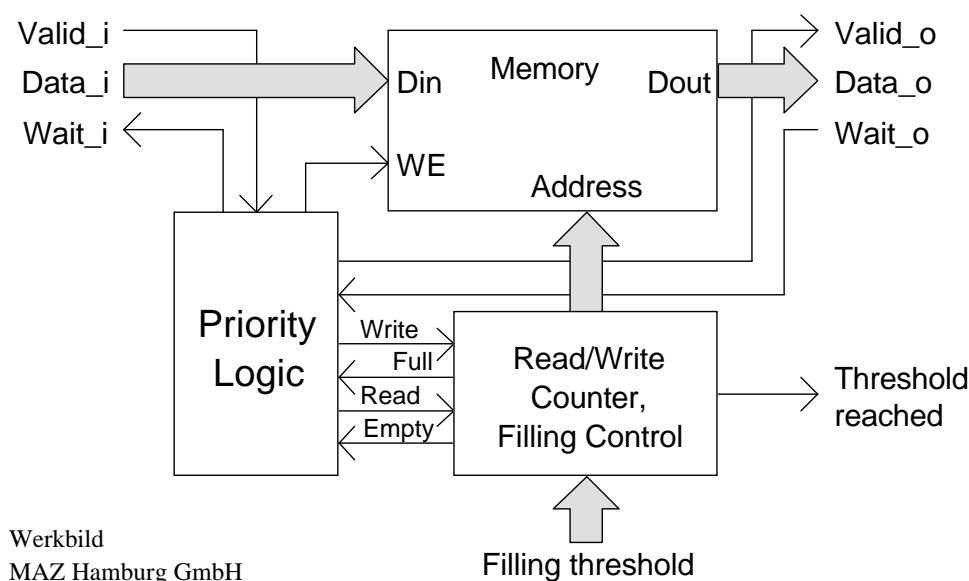


Abbildung 4: Datenfluß FIFO Speicher

Bild 4 zeigt das Blockschaltbild eines Datenfluß-FIFOs. Ein Speicherblock wird aus entsprechend konfigurierten CLBs realisiert und kann alternativ gelesen oder geschrieben werden. Eine Steuereinheit beinhaltet den Schreib- und Lesezähler sowie Logik zur Überwachung des FIFO-Füllstands. Eine Prioritätslogik löst schließlich Zugriffskonflikte bei gleichzeitiger Lese- und Schreibanforderung mithilfe des elastischen Datenflußprotokolls auf. Zwei alternative Prioritätslogiken geben dem Eingang oder dem Ausgang den Vorrang bei Konflikten. Angepaßt an den Datenfluß im SE-XIL finden beide FIFO-Typen Verwendung und gewährleisten jeweils benötigte Zugriffe auf Zelldaten.

Die Ausgänge werden von einem Funktionsblock gesteuert, der das Ausgangs-FIFO liest und nach einer Wort/Byte Konversion bei jeder Zelle den VCI auswertet. Gemäß einer statischen Zuordnung wird eine Zelle über selektierte Ausgänge ausgegeben, die übrigen Ausgänge geben leere Zellen (*Idle* Zellen) synchron aus. An dieser Stelle findet somit beim SE-XIL-Element das Routing statt.

Die Spezifikation des SE-XIL-Elements erfolgte in VHDL, wobei Teile der ASIC-Spezifikation übernommen wurden. Die Synthese wurde mit Synopsis durchgeführt. Die aktuelle Implementierung belegt 70 – 80% der CLB's in einem XC4013 LCA. Der Systemtakt mußte gegenüber der ASIC-Version von 20 MHz auf 16 MHz verringert werden.

Unter Verwendung des SE-XIL-Elements und weiterer LCA-basierter Prototypbausteine konnte die erste Version einer ATM-Vermittlungseinheit realisiert und frühzeitig an die Firma Wenzel, Elmshorn ausgeliefert werden. Die Einheit ist Bestandteil einer neuen ISDN Vermittlungsanlage für die Deutsche Bundesbahn, die mittlerweile im Bahnhof Marktredwitz in Betrieb ist.

4 Zusammenfassung

Die Verwendung von LCAs als VLSI-Ersatz mit eingeschränkter Funktionalität ermöglichte beim Entwurf eines ATM-Schalterbausteins eine frühe Verifikation des Gesamtsystems und eine frühzeitige Auslieferung einer ATM-Vermittlungseinheit an den Kunden. Damit wurde der zeitliche Ablauf des Systementwurfs weniger abhängig von Verzögerungen bei der ASIC-Entwicklung und Fertigung.

Die Integration komplexer VLSI-Subfunktionen in ein LCA wurde möglich durch den gewählten Datenflußansatz, durch sorgfältige Auswahl und Analyse benötigter Datenpfade und durch eine gezielte Synthese, mit der die LCA-Ressourcen effizient ausgenutzt werden.

Literatur

- [Sut89] Ivan E. Sutherland. Micropipelines. *Communications of the ACM*, Vol. 32, Nr. 6, June 1989.
- [MAZ] MAZ GmbH. Distributed ATM Switch DAS, Switchelement (SE) DAS 340, Preliminary. Hamburg, 1994.
- [Lan94b] B. Lang. Self Arbitrating Elements for Modelling Systolic Dataflow in Field Programmable Gate Arrays. In *Proceedings GI/ITG Workshop Anwenderprogrammierbare Schaltungen*, Karlsruhe, 1994.
Verfügbar via Internet: <ftp://www.ti1.tu-harburg.de/pub/papers/la:ka.ps>