

Bernd Däne / Falk Berger / Alexander Pacholik

# **Ein eingebettetes Multiprozessorsystem für die Steuerung von Nanomess- und Positioniermaschinen**

## **ABSTRAKT**

Der Beitrag beschreibt ein Konzept und eine experimentelle Realisierung für ein eingebettetes Mehrprozessorsystem mit Digitalen Signalprozessoren. Das System weist in der Kommunikationsstruktur einige Besonderheiten auf. Insbesondere wurde eine Broadcast-Funktion zur gleichzeitigen Übertragung identischer Daten zu mehreren Teilnehmern realisiert.

## **1. Einführung**

In Nanomess- und Positioniermaschinen muss eine schnelle und präzise Positionsregelung erfolgen. Die dafür benötigten Regelalgorithmen werden in schnellen eingebetteten Systemen implementiert, welche häufig mit Digitalen Signalprozessoren (DSP) realisiert werden. Aufgrund der Anzahl von Bewegungsachsen, welche in die Regelung einbezogen sind, liegt stets eine mehrkanalige Regelung vor. Ein wichtiges Entwurfsziel ist die Erzielung einer hohen Abtastrate, da hiervon die erreichbaren dynamischen Eigenschaften der Nanomess- und Positioniermaschinen abhängen. Die benötigte Verarbeitungsleistung kann durch den Einsatz eines Multiprozessorsystems erzielt werden. Als Engpass tritt dabei jedoch der Datentransport zwischen den Prozessoren auf. In diesem Beitrag wird ein Multi-DSP-System vorgestellt, welches diesem Umstand durch eine problemangepasste Gestaltung des Kommunikationssystems Rechnung trägt.

## **2. Grundstruktur des Multi-DSP-Systems**

Das beschriebene Multi-DSP-System ist für sechs Digitale Signalprozessoren vom Typ TMS320C6701 [1] des Herstellers Texas Instruments Inc. konzipiert und wurde im Zusammenhang mit einem Gemeinschaftsprojekt zu Nanomess- und Positioniermaschinen [2] entworfen. Eine detaillierte Beschreibung des Multi-DSP-Systems ist in [3] enthalten.

Abb. 1 zeigt die Grundstruktur des Systems und verdeutlicht die Arbeitsteilung zwischen den Prozessoren. Einer der Prozessoren agiert als Master und organisiert alle Datentransporte innerhalb des

Systems. Vier gleichberechtigte Slaves übernehmen die eigentliche Rechenlast bei der Berechnung der Regelalgorithmen. Ein weiterer Prozessor (Comm) ist für die Kommunikation mit einem übergeordneten Leitrechner reserviert, wofür ein schnelles Standardinterface vom Typ USB 2.0 zum Einsatz kommt.

Für die Verbindungen werden Interfaces genutzt, die bei den benutzten DSPs on-chip verfügbar sind. Das External Memory Interface (EMIF) realisiert eine Parallelbus-Schnittstelle, an welcher der betreffende DSP als aktiver Busmaster auftritt. Das Hostport-Interface (HPI) erlaubt dagegen den Zugriff eines externen Busmasters auf die internen Speicherkomponenten des betreffenden DSP, wobei sich dieser ähnlich einer Speicherbaugruppe, also passiv, verhält.

Durch geeignete Kopplung dieser Interfaces werden die Kommunikationsverbindungen realisiert, wobei vom Master aus gesehen zwei unabhängige Zugriffswege zum Kommunikationsprozessor und zu den Slaves existieren. Darüber hinaus ist eine Schnittstelle zum Zugriff auf die Ein- und Ausgabeports der Sensor- und Aktorbaugruppen vorhanden (Port-IO). Auf Grund der kritischen Geschwindigkeitsanforderungen und der relativ großen Leiterzuglängen erwiesen sich widerstandsangepasste, impedanzkontrollierte Leiterzüge und auch Busstreiber als notwendig [3]. Gewählt wurde eine beidseitige Parallelterminierung mit geteiltem Abschlusswiderstand [4].

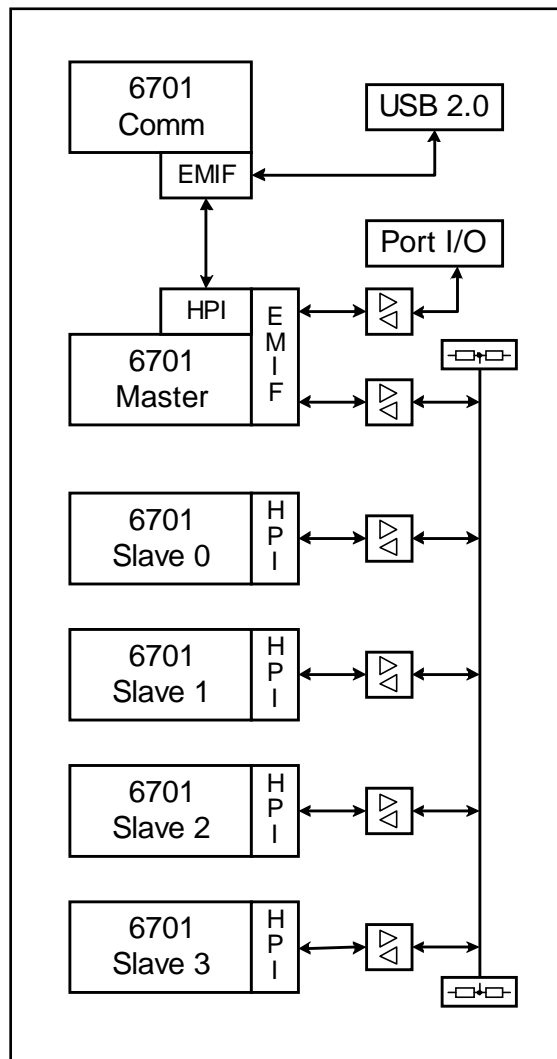


Abb. 1: Grundstruktur

### 3. Problemangepasste Gestaltung des Kommunikationssystems

Die Verbindung zwischen dem Master und dem Kommunikationsprozessor ist als Punkt-zu-Punkt-Verbindung zwischen einem EMIF und einem HPI ausgeführt [5] und weist diesbezüglich keine Besonderheiten auf. Da der Master hier als passiver Teilnehmer auftritt, wird sein Prozessorkern von dieser Kommunikationslast nicht wesentlich beeinflusst.

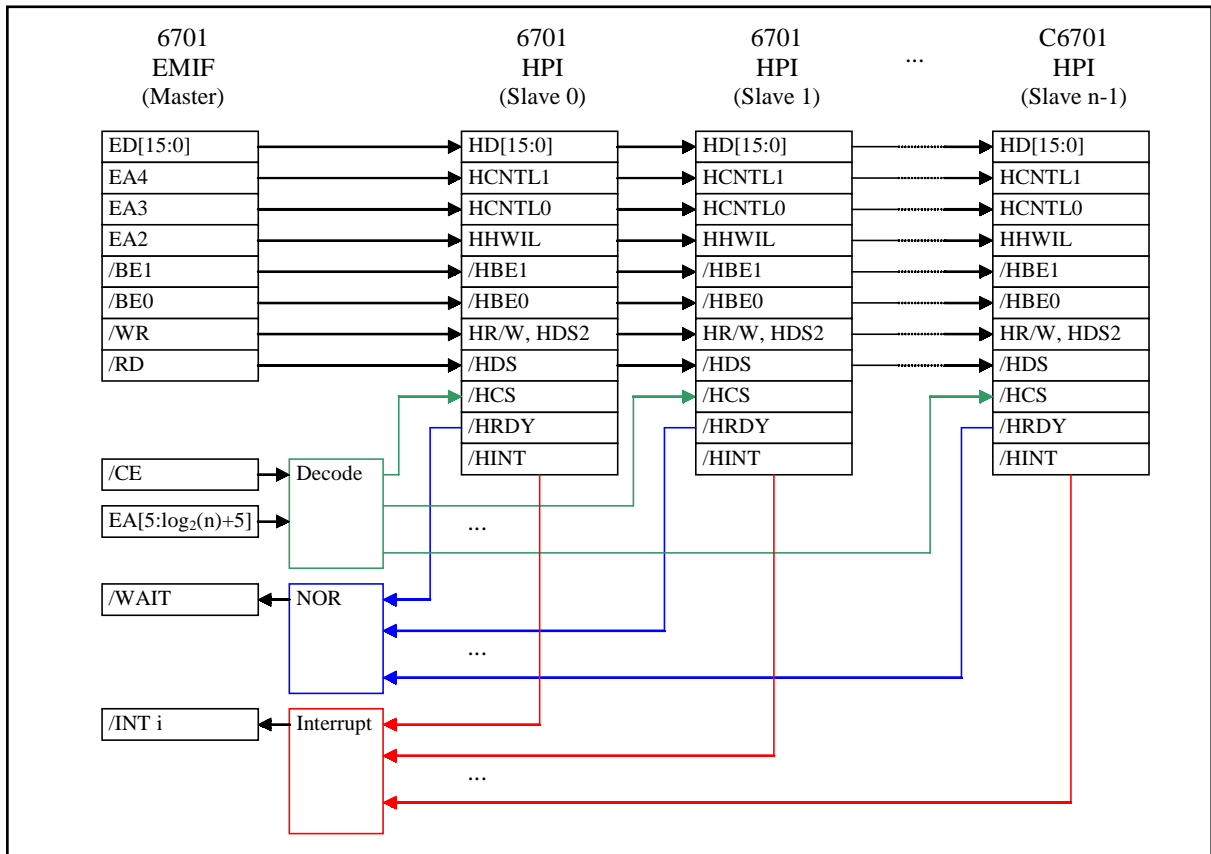


Abb. 2: Verknüpfung von Steuersignalen für den broadcastfähigen Bus

Die vier Slaves sind jedoch mittels einer Busstruktur mit dem Masterprozessor verbunden. Dabei tritt letzterer als Busmaster auf, während sich alle Slaves als passive Teilnehmer verhalten. Dadurch ist es möglich, auf sämtlichen Slaves Endlosschleifen mit den Berechnungsalgorithmen laufen zu lassen, ohne zusätzliche Kommunikationskomponenten mit entsprechendem Overhead (z.B. durch Kontextwechsel) implementieren zu müssen. In dieser Konfiguration kann in den Slaves ein Maximum an Verarbeitungsdurchsatz erreicht werden.

Die Busstruktur für die Verbindung des Masterprozessors mit den Slaves musste speziell entworfen werden, da das verwendete DSP-System keine eigenen Vorkehrungen für den Aufbau solcher Multiprozessorsysteme anbietet. Da die Slaves aus Sicht des Masters in der Art von Speicherbereichen erscheinen, werden sie durch Adressbereiche analog der Adressierung von Speicherbaugruppen unterschieden. Um Zeitvorteile bei der Ansteuerung der Treiberbausteine zu erreichen, wurden für das Schreiben und für das Lesen jeweils unterschiedliche Adressbereiche festgelegt.

Als besondere Funktion wurde zusätzlich zum individuellen Ansprechen der Slaves eine Broadcast-Funktion realisiert. Unter Nutzung eines separaten Adressbereiches kann der Master gleichzeitige Schreibvorgänge auf alle Slaves ausführen. Das führt zu starker Einsparung an Kommunikations-

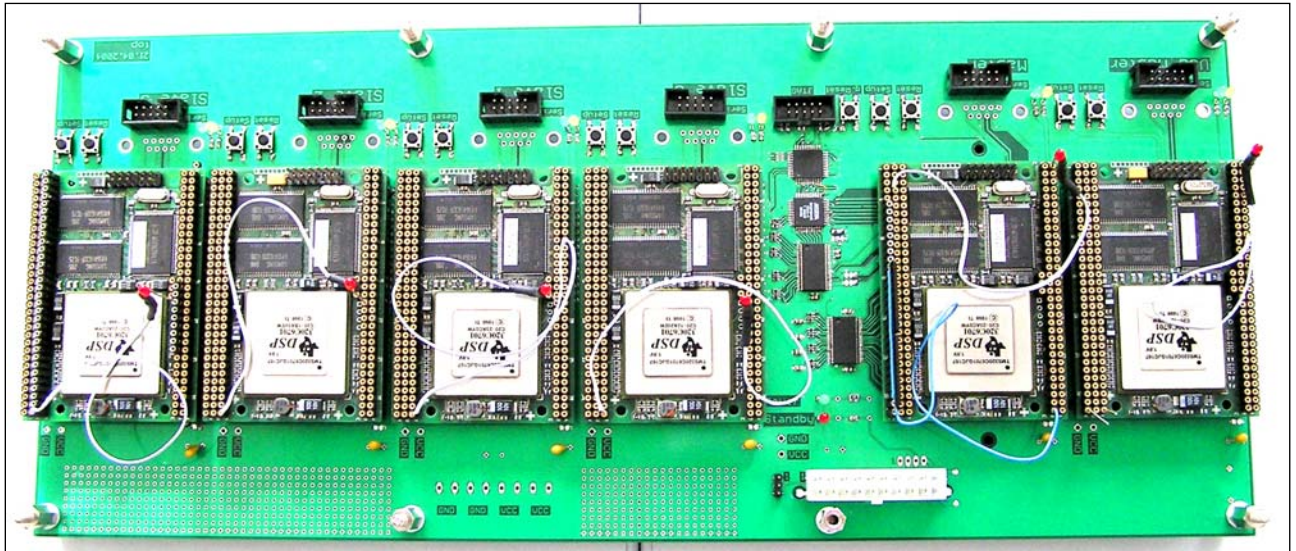


Abb. 3: Ansicht der Leiterplatte (Module v. r. n. l.: Comm, Master, Slaves 0 bis 3)

zeit, falls alle Slaves mit identischen Daten zu versorgen sind. In der vorliegenden Applikation ist dieser Fall von großer Relevanz, da in jedem Takt die kompletten Messdaten der Sensorik aufgrund der Verkopplung der Regelkreise von allen Reglern benötigt werden. Die berechneten Regelgrößen werden dagegen durch den Master einzeln von den Slaves abgeholt.

Abb. 2 zeigt schematisch die Verknüpfung der relevanten Steuersignale. Die Auswahlssignale für die Slaves werden durch eine gewöhnliche Adressdekodierung aus der vom Master ausgegebenen Speicheradresse erzeugt. Tritt in einem Slave während eines Zugriffs des Masters eine Verzögerung auf (z.B. durch einen internen Zugriffskonflikt), wird mit Hilfe eines Wait-Signals der Zyklus entsprechend verlängert. Dieser Fall hat allerdings eine geringe Wahrscheinlichkeit, so dass er in der gesamten Zeitbilanz nicht stört. Für den Broadcast-Modus muss eine entsprechende Verknüpfung der Wartesignale der einzelnen Slaves gebildet werden, so dass die Länge des Buszyklus stets durch den

Richtung und Modus	Datenrate (MByte/s)
Lesen von einem Slave (Einzelzugriffe)	6,1
Lesen von einem Slave (Burst-Mode)	13,3
Schreiben auf einen Slave (Burst-Mode)	14,5
Broadcast zu allen Slaves (Burst-Mode)	14,5

Tab. 1: Gemessene Übertragungsraten

langsamsten Teilnehmer bestimmt wird. Ein zusätzliches Interrupt-Signal kann bei Notwendigkeit zur Signalisierung besonderer Zustände der Slaves zum Master dienen, wobei aufgrund der begrenzten Anzahl externer Interrupteingänge eine Zusammenfassung zu einem Sammelinterrupt erfolgen muss.

#### **4. Resultate und Zusammenfassung**

Das beschriebene Rechnersystem wurde unter Laborbedingungen realisiert und experimentell untersucht. Abb. 3 zeigt eine Ansicht der realisierten Leiterplatte. Die Prozessoren befinden sich auf Piggyback-Modulen des Typs D.Module.C6701 [6] vom Hersteller D.SignT GbR. Für den Funktionstest wurde Kommunikationssoftware mit an den MPI-Standard [7] angelehnten Elementarprozeduren implementiert. Durch Nutzung von DMA-Funktionen [1] sowie durch Unterstützung des Burst-Modus des HPI [5] konnte die Effektivität optimiert werden. Tab. 1 zeigt die erreichten Übertragungsraten für die Datenübertragung zwischen dem Master und den Slaves. Es zeigt sich, dass der Broadcast-Modus dasselbe Zeitverhalten aufweist wie das Schreiben zu einem einzelnen Slave. Da hierbei die Daten aber gleichzeitig zu allen Slaves transportiert werden, lassen sich erhebliche Zeitvorteile erreichen.

#### **5. Ausblick**

Künftige Entwicklungen betreffen die algorithmische Optimierung mit Bezug zur Zielapplikation, d.h. die Anpassung der Kommunikationsprozeduren an die Spezifik der eingesetzten Applikationssoftware. Weiterhin soll eine im Konzept bereits vorbereitete Erweiterung auf eine skalierbare Anzahl von Prozessoren ausgeführt werden. Die entworfene Hardware soll für die Verwendung anderer Prozessormodelle innerhalb der verwendeten Prozessorfamilie adaptiert werden. Von Interesse ist hier z.B. der Typ TMS320C6713, der eine verbesserte Verarbeitungsleistung und insbesondere eine flexiblere DMA-Funktion bietet, allerdings auch einige Unterschiede im Bereich der HPI-Bedienung aufweist. Schließlich soll das schnelle multikanalfähige serielle Interface der benutzten DSPs (McBSP, [1]) als zusätzlicher Kommunikationsweg innerhalb des Systems nutzbar gemacht werden, da es bei geringem Bedarf an Verbindungsleitungen eine relativ schnelle Datenübertragung bietet und das gleichberechtigte Versenden und Empfangen von Nachrichten durch alle beteiligten Prozessoren ermöglicht.

*Diese Arbeit wurde gefördert von der Deutschen Forschungsgemeinschaft (DFG) im Rahmen des Sonderforschungsbereiches SFB 622.*

**Referenzen:**

- [1] Naim Dahnoun: Digital Signal Processing Implementation Using the TMS320C6000 DSP Platform. Prentice Hall 2000, ISBN 0-201-61916-4.
- [2] Tino Hausotte: Nanopositionier- und Nanomessmaschine. ISLE Verlag 2002, ISBN 3-932633-71-7.
- [3] Alexander Pacholik: Konzeption und Realisierung eines Multi-DSP-Systems und seiner Schnittstellen. Diplomarbeit TU Ilmenau (IA) 2004.
- [4] Howard Johnson: "High-Speed Digital Design". Prentice Hall 1993. ISBN 0-13-395724-1.
- [5] Zoran Nikolic: TMS320C6000 EMIF to TMS320C6000 Host Port Interface. Application Report SPRA536B, Texas Instruments Inc., 2003.
- [6] D.Module.C6701. Technical Data Sheet. Revision 2.1, June 2002. Firmenschrift D.SignT GbR Kerken, 2002.
- [7] Message Passing Interface Forum: MPI, a Message-Passing Interface Standard. University of Tennessee, Knoxville, Tennessee, 1994.

**Autorenangaben:**

Dr.-Ing. Bernd Däne

Dipl.-Inf. Falk Berger

Dipl.-Ing. Alexander Pacholik

TU Ilmenau, PSF 100565

98684 Ilmenau, Germany

Tel.: +49-3677-69-2825

Fax: +49-3677-69-1614

E-Mail: (bernd.daene|falk.berger|alexander.pacholik)@tu-ilmenau.de