Hochgenaue Uhrensynchronisation über LANs

(erscheint in *Elektronikschau*, 1997)

ULRICH SCHMID*
Technische Universität Wien
Institut für Automation
Treitlstraße 3, A-1040 Wien
Email: s@auto.tuwien.ac.at
Telefon: ++43-1-58801-8189, Fax: ++43-1-5863260

16. September 1997

Abstract

Eine im Rahmen des wissenschaftlichen Projekts SynUTC[1] an der TU-Wien entwickelte Hard- und Software erlaubt es, die Knoten eines netzwerkgekoppelten Mikroprozessorsystems mit im μs-Bereich zueinander und auf Weltzeit synchronisierten Uhren auszurüsten. Wohldefinierte theoretische Grundlagen, exzellente Fehlertoleranzeigenschaften sowie weitgehende Flexibilität bezüglich der zugrundeliegenden Hardware und Netzwerktechnologie sind weitere Charakteristika der entwickelten *enabling technology*, mit deren Hilfe völlig neue Anwendungsgebiete erschlossen werden können.

[0]Der Autor ist Dozent am Institut für Automation der TU-Wien und einer der Preisträger des österreichischen START-Preises 1996.

[1]Das gegenständliche Projekt SynUTC ist ein Kooperationsprojekt zwischen dem Institut für Automation (Ulrich Schmid) und dem Institut für Computertechnik (Dietmar Loy) der TU-Wien, das vom Österreichischen Fonds zur Förderung der Wissenschaftlichen Forschung (FWF) unter Projektnummer P10244-ÖMA und vom BMWV im Rahmen des START-Programms Nr. Y41-MAT gefördert wurde. Nähere Informationen sind unter http://www.auto.tuwien.ac.at/Projects/SynUTC/ zu finden.

1. Einleitung

"Mangelnde Zusammenarbeit zwischen Universität und Industrie" — eine Schlagzeile, deren Wahrheitsgehalt und Aktualität neben wirtschaftspolitisch interessierten Zeitungslesern auch die unmittelbar Betroffenen allmählich zu realisieren beginnen. Die immer stärker werdende Tendenz, wissenschaftliche Institute zu Entwicklungsabteilungen für Industrieaufträge umzufunktionieren, ist allerdings eher eine besorgniserregende Fehlentwicklung denn eine zukunftsträchtige Lösung, da die Schaffung von Entwicklungskompetenz mittelfristig zu Lasten der Forschungskompetenz gehen muß.

In der vom Autor geleiteten Arbeitsgruppe für verteilte Systeme am Institut für Automation der TU-Wien wird daher ganz bewußt versucht, einen anderen Weg hinaus aus dem "elfenbeinernen Turm" einzuschlagen: Unbeschadet ihres anspruchsvollen wissenschaftlichen "Innenlebens" zielen die dort abgewickelten, *selbstgewählten* Forschungsprojekte primär darauf, eine möglichst optimale Lösung eines *anwendungsnahen Problems* zu erarbeiten und deren tatsächliche Tauglichkeit mit Hilfe einer Prototyp-Implementierung zu demonstrieren. Erst durch einen evaluierbaren Prototypen wird es für den potentiellen Anwender nämlich möglich, die Brauchbarkeit einer Lösung und den Aufwand für die konkrete Adaptierung realistisch einzuschätzen; dies allein auf Basis noch so guter wissenschaftlicher Publikationen bewerkstelligen zu wollen, ist erfahrungsgemäß so gut wie aussichtslos.

Das gegenständliche Projekt SynUTC (*Synchronized Universal Time Coordinated for Distributed Fault-Tolerant Real-Time Systems*) fällt in diese Kategorie von Forschungsprojekten. Es ist konkret der Fragestellung gewidmet, wie —ohne dedizierte GPS-Empfänger oder Zusatzverkabelung heranzuziehen— die Knoten eines netzwerkgekoppelten, fehlertoleranten Rechnersystems mit hochgenau untereinander und auf Weltzeit synchronisierten lokalen Uhren ausgerüstet werden können. Unbeschadet der komplexen theoretischen und konzeptuellen Grundlagen, die in zahlreichen wissenschaftlichen Arbeiten publiziert wurden, ist das eigentliche Endergebnis des Projekts eine "vorzeigbare" Prototyp-Implementierung von Hard- und Software für ein derartiges System (siehe Bild 3).

Der vorliegende Artikel soll nun dazu beitragen, die erarbeiteten Projektergebnisse bekanntzumachen und so konkrete Anwendungsmöglichkeiten zu erschließen. Die Anwendungsnähe eines Forschungsprojekts impliziert ja keineswegs die Kenntnis aller möglichen Einsatzgebiete. Tatsächlich ist im Falle von SynUTC bisher nur eine einzige — allerdings bereits patentierte— konkrete Anwendung im Bereich Kabelfehler-Ortung bekannt, in der die erzielbare Synchronisationsgenauigkeit von $< 1\ \mu s$ wirklich benötigt wird. Aufgrund einer weiteren, sich abzeichnenden Verwendungsmöglichkeit im Bereich (Multimedia-)Datenkommunikation erscheint es jedoch berechtigt, auf den Sog-Effekt von *enabling technology* auf neuartige Anwendungen zu vertrauen.

Da die Forschungsarbeiten in SynUTC ohne Bindung an einen Industriepartner durchgeführt wurden, ist es möglich und beabsichtigt, die Ergebnisse im Prinzip jedem Interessenten zur Weiterverwertung zur Verfügung zu stellen (wobei auf Grund der Finanzierung mit öffentlich-rechtlichen Mitteln österreichische Firmen natürlich besonders willkommen sind). Alle potentiellen Anwender sind also herzlich eingeladen, mit dem Autor diesbezüglich Kontakt aufzunehmen.

2. Features

Anders als bei herkömmlichen Verfahren basierend auf einer zentralen Uhr ist jeder SynUTC-Knoten mit einer eigenen lokalen Uhr ausgerüstet, die allein durch den Austausch von Synchronisationsnachrichten über das verbindende Datennetzwerk mit den anderen Uhren im System auf μs genau synchronisiert wird. Der (optionale) Bezug zur Weltzeit wird dadurch hergestellt, daß an mindestens einem der Rechner ein GPS Satellitenempfänger mit 1 pps-Zeitausgang angeschlossen sein muß. Im Gegensatz zu existierenden Verfahren ist es hier jedoch nicht notwendig, jeden Knoten mit einem dedizierten GPS-Empfänger auszurüsten oder Zeitinformation über zusätzliche Verbindungsleitungen zu verteilen. Das Bild 1 zeigt schematisch die wichtigsten Komponenten eines Systems mit zwei Knoten.

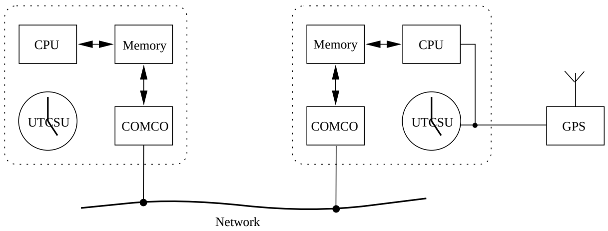

Figure 1: *Zwei Knoten eines verteilten Systems*

Die im Rahmen von SynUTC entwickelte Hardware/Software basiert auf theoretisch wohlfundierten Grundlagen und kann durch umfangreiche mathematisch-formale Analysen den Anspruch erheben, die folgenden Leistungsmerkmale —von Implementierungsfehlern abgesehen— auch garantiert zu erbringen:

- Überbrückung von LAN-Distanzen unter alleiniger Verwendung des vorhandenen Datennetzwerks. Die Eignung eines bestimmten Netzwerks/Communication-Controllers hängt dabei nicht vom deterministischen Anteil der Nachrichtenverzögerungszeit δ, der automatisch gemessen wird, ab, sondern lediglich vom maximalen *Delay-Jitter* $\delta_{\max} - \delta_{\min}$.

- Hohe Auflösung (Granularität) der lokalen Uhren von $2^{-24} \approx 60$ ns.

- Hohe Synchronisationsgenauigkeit der lokalen Uhren zueinander (genannt *Precision*), die linear vom Delay-Jitter des zugrundeliegenden Netzwerks und der verwendeten Taktfrequenz der lokalen Uhren abhängt; mit Ethernet und 10–20 MHz Taktfrequenz ist etwa 1 μs (worst-case, mit wesentlich besserem average-case) zu erreichen.

- Im Normalfall hohe Synchronisationsgenauigkeit der lokalen Uhren zur Weltzeit (genannt *Accuracy*), auch bei Knoten ohne[2] GPS-Empfänger. Darüberhinaus Bereitstellung einer zuverlässigen worst-case Qualitätsinformation (= maximale Abweichung) der momentanen Accuracy.

- Exzellente Fehlertoleranz-Eigenschaften, v.a.

  - Uhrensynchronisation wird durch fehlerhafte Rechner/Uhren nicht beeinträchtigt,
  - (längerer) Ausfall oder Fehlverhalten von GPS-Empfängern verschlechtert lediglich die Accuracy, nicht aber die Precision,
  - mehrere Selbsttest- und Datensicherungsmechanismen.

- Möglichkeit des Aufbaus redundanter Systemarchitekturen, v.a.

  - redundante (mehrfache) Netzwerkverbindungen,
  - redundante Ausführung der lokalen Uhren,
  - Anschluß mehrerer GPS-Empfänger.

Für die auf SynUTC aufbauenden Anwendungen sind —neben der Möglichkeit, jederzeit die aktuelle Uhrzeit per Software auslesen zu können— vor allem die vielfältigen Zeitstempel-Funktionen interessant: Zum Zeitpunkt der Zustandsänderung eines digitalen Eingangs wird der Stand der lokalen Uhr in ein dediziertes Register gespeichert, wodurch ein solcher Zeitstempel (zeitlich relativ unkritisch, aber natürlich noch vor der nächsten Zustandsänderung) per Software ausgelesen und weiterverarbeitet werden kann. Weiters stehen Möglichkeiten zur Auslösung von Interrupts bzw. eines Pulses an einem digitalen Ausgang bei Erreichen eines bestimmten Zeitpunkts zur Verfügung.

3. Realisierung

Die gesamte im Rahmen von SynUTC entwickelte Hard- und Software basiert auf dem sogenannten *interval-based paradigm*, in dem die Zeit t auf einem Knoten p nicht wie üblich durch einen punktförmigen *clock value* $C_p(t)$, sondern durch ein *accuracy interval* $\boldsymbol{A}_p(t) = [C_p(t) - \alpha_p^-(t), C_p(t) + \alpha_p^+(t)]$ repräsentiert wird. Aufgabe der Uhrensynchronisation ist es, die *accuracy intervals* so zu steuern, daß für alle fehlerfreien Knoten p und q folgendes garantiert werden kann:

(P) *Precision*: $|C_p(t) - C_q(t)| \leq \pi_{\max}$ für ein vorgegebenes $\pi_{\max} \geq 0$,

(A) *Accuracy*: $t \in \boldsymbol{A}_p(t)$, wobei das Genauigkeitsinterval $[-\alpha_p^-, \alpha_p^+]$ minimal sein soll.

Praktisch wird dies so gelöst, daß alle Knoten periodisch ihr *accuracy interval* an alle anderen Knoten versenden und auf Basis der eingegangenen Informationen einen Korrekturwert für die eigene Uhr berechnen.

Jeder Rechner muß dazu über folgende, im Bild 1 schematisch dargestellte Hardware-Komponenten verfügen:

- *CPU* für die Ausführung der Synchronisationsalgorithmen, die jedoch ohne Schwierigkeiten parallel zu anderen Tasks exekutiert werden können.

- *DMA-fähiger Communication-Controller* für ein geeignetes Netzwerk, der das Senden/Empfangen von Datenpaketen aus/in den Speicher ohne Mithilfe der CPU erledigen kann.

[2]Die Accuracy von Knoten mit direktem GPS-Anschluß ist natürlich besser. Übrigens existiert für "isolierte" Konfigurationen (ohne Netzwerk) auch die Möglichkeit, die lokale Uhr mit dem 1 pps-Puls eines direkt angeschlossenen GPS-Empfängers phase-locked zu synchronisieren.

- *Memory* zum Austausch von Datenpaketen zwischen CPU und Netzwerk-Controller.
- *UTCSU-ASIC* (siehe Bild 2), das den primären Hardware-Support für die Uhrensynchronisation (v.a. die lokale Uhr und diverse Zeitstempelmechanismen) bereitstellt. Durch den Einsatz von Rate-Synchronisation genügt ein TCXO mit beliebiger Frequenz im Bereich 1...20 MHz als Taktgeber.

In der im Rahmen von SynUTC entwickelten Prototyp-Hardware (siehe Bild 3) sind UTCSU-ASIC, Speicher und diverse Interface-Logik in Form eines einzelnen *Network Time Interface* (NTI) M-Moduls [MM96] realisiert. Die NTI kann daher im Prinzip zusammen mit jedem CPU-Modul mit MA-Interface verwendet werden.

Die Prototyp-Software wurde konkret für ein Ethernet-gekoppeltes System bestehend aus 4 Motorola MVME-162 CPU's (68040 + i82596) geschrieben, die —aus Kostengründen— jeweils 4 NTIs gleichzeitig bedienen. Die Kopplung zwischen einer MVME-162 und ihren NTIs erfolgt über den VME-Bus und eine als passiver "VME/MA-Buskoppler" fungierende AcQ i6040+EXT VME-CPU. In näherer Zukunft soll die Software, die übrigens als Device-Driver für ISI's Multiprozessor-Echtzeitbetriebssystem pSOS^{+m} konzipiert wurde, auch direkt auf die i6040 (68040 + 68EN360) portiert werden.

Das auf diese Weise realisierte verteilte System mit 16 Knoten wird primär zur experimentellen Verifikation der theoretischen Resultate und natürlich für Demonstrationszwecken herangezogen. Es wird voraussichtlich Ende dieses Jahres zur Verfügung stehen.

References

[HSS97] M. Horauer, U. Schmid, K. Schossmaier. *NTI: A Network Time Interface M-Module for High-Accuracy Clock Synchronization*, Manuskript (eingereicht zur Veröffentlichung), 1997.

[MM96] Manufacturers and Users of M-Modules. *M-Module Specification*, April 1996, MUMM e.V., Simon-Schöffel-Straße 21, D-90427 Nürnberg, Tel. ++49-911-3067170.

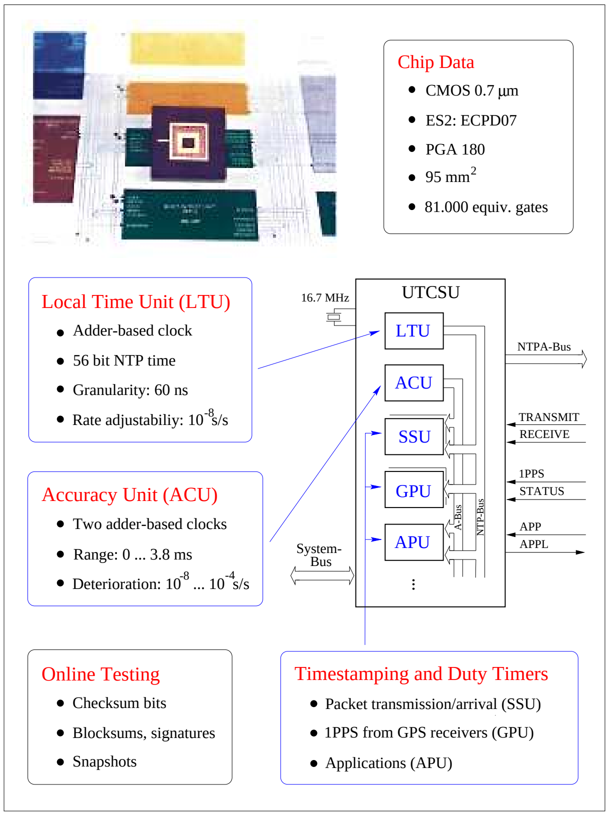

Figure 2: *Features des entwickelten UTCSU-ASICs*

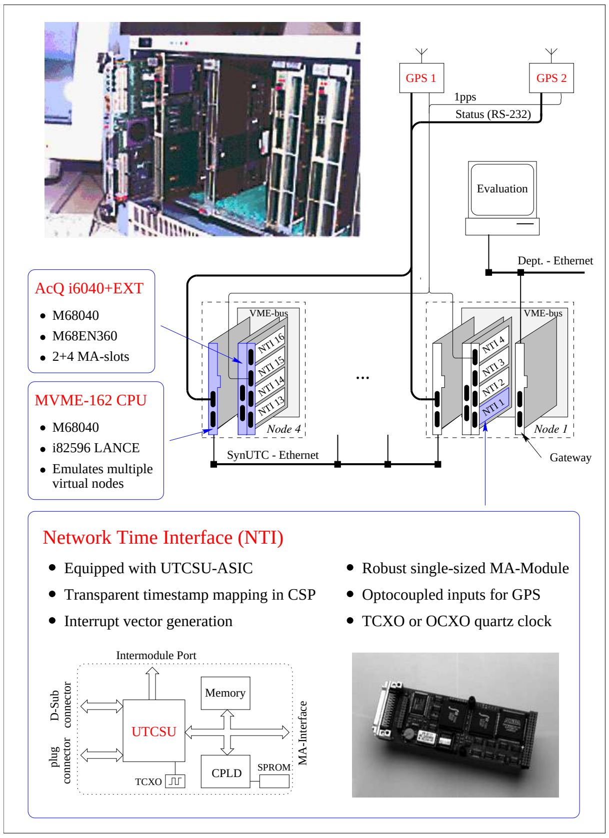

Figure 3: *Übersicht Prototyp-Implementierung der Hardware*