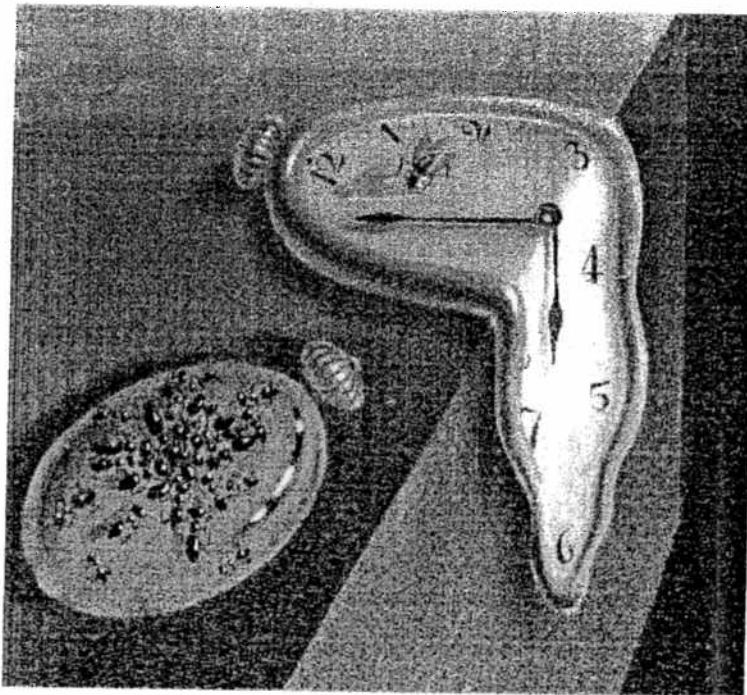


Projektbericht Nr. 183/1–36  
April 1993

**Zusatzantrag zum FWF–Projekt VTA**  
*Ulrich Schmid*



Ausschnitt aus: Salvador Dalí, "Die Beständigkeit der Erinnerung"

# **Zusatzantrag**

## **zum Projekt**

# **Versatile Timing Analyzer**

## **(P8390-PHY)**

Ulrich Schmid, 22.4.1993

Beilagen:    3 Angebote/Unterlagen zu den Geräten  
                 4 Papers zum Projekt (HS1, S2, SK2, St2)  
                 1 Datenblatt CSU

Der vorliegende Zusatzantrag zum Projekt VTA stellt eine umgearbeitete und ergänzte Fassung eines bereits im Jahre 1992 gestellten – und abgelehnten – Zusatzantrages dar. Leider wurden mir die kritischen Anmerkungen der Fachgutachter zu dieser Erstfassung erst nach der erfolgten Ablehnung zur Kenntnis gebracht, sodaß ich nicht eher darauf reagieren konnte. Tatsächlich war es mir damals nicht einmal klar, daß der Zusatzantrag einer *wissenschaftlichen* Evaluation unterzogen werden würde, da der FWF für mein Projekt im Rahmen der von der Oesterreichischen Nationalbank geförderten Kategorie der *wirtschaftsnahen Projekte* eine umfangreiche Förderung (d.h., Refundierung der FWF-Förderungsmittel) erhalten hatte. In Anbetracht dessen erschien es mir naheliegend, bei der Formulierung des Zusatzantrages eher die Entwicklungsaspekte als die (nicht minder vorhandene) "Wissenschaftlichkeit" hervorzuheben. In diesem Zusammenhang sollte noch erwähnt werden, daß das Forschungsprojekt VTA – wie wohl sehr viele Informatikprojekte – ganz klar in die Kategorie *oriented basic research* einzuordnen ist.

Diese Neufassung des Zusatzantrages trägt diesem Umstand Rechnung, soll also die wissenschaftliche Seite der beantragten Zusätze detailliert begründen. Darüberhinaus berücksichtigt er auch einige neue (und wesentliche) Umstände, die zum Zeitpunkt der Erstfassung noch nicht vorlagen.

In der Anlage finden sich einige Arbeiten, die im Rahmen der Projektarbeit entstanden sind. Sie sollen es den Fachgutachtern ermöglichen, sich – über die noch folgenden Darstellungen hinaus – ein Bild von der bisher geleisteten Projektarbeit zu machen. Ein Verzeichnis aller bisherigen Arbeiten ist im Anhang A zu finden.

## 1. Status des Projektes

Das Projekt VTA (*Versatile Timing Analyzer*, P8390-PHY) beschäftigt sich mit der Entwicklung der Grundlagen sowie des Prototyps eines flexiblen Monitoring-Systems zur Timing-Analyse verteilter Echtzeitsysteme. Aufgrund der bisherigen Erfahrungen mit dem seit 1.8.1991 laufenden, 2-jährigen Forschungsprojekt erscheint es nun ratsam und möglich, einige (zum Teil bereits von den FWF-Gutachtern angeregte) Erweiterungen, die aus pragmatischen Gründen zunächst nicht vorgesehen waren, zu realisieren. Die Voraussetzung dafür ist allerdings die Bereitstellung zusätzlicher Förderungsmittel sowie eine Verlängerung der Projektdauer.

Einleitend ist festzuhalten, daß die bisherigen Erfolge des Forschungsprojektes sehr ermutigend sind. So wurden unter anderem – trotz der kurzen Laufzeit – mehrere diesbezügliche Arbeiten bei internationalen wissenschaftlichen Konferenzen angenommen. Auf einigen dieser Tagungen, wie z.B. dem 14-tägigen *NATO Advanced Studies Institute on Real-Time Computing* in St. Martin, die sehr viel Spielraum für persönliche Diskussionen ließen, zeigte sich deutlich, daß viele Wissenschaftler an einem Werkzeug wie dem VTA interessiert wären – daß aber gegenwärtig niemand (außer uns) daran arbeitet. Darüberhinaus beweisen auch unsere Industriekontakte – nicht zuletzt durch die erfolgte Förderung durch die

OeNB im Rahmen der wirtschaftsnahen Projekte – sehr deutlich das "kommerzielle" Interesse an dem VTA, sodaß das mittelfristig geplante, fortführende FFF-Projekt zur Entwicklung eines "kommerziell" verwendbaren Systems sicherlich nicht utopisch bleiben wird.

Das gesamte Forschungsprojekt ist in insgesamt drei voneinander relativ unabhängige Teilgebiete zu unterteilen (siehe [S2]), und zwar

**(A) Primitive Event Management**

Dieser Bereich umfaßt alle Fragestellungen, die mit der Erfassung der primitiven Ereignisse (Events) in einem einzelnen Node des zu beobachtenden verteilten Echtzeitsystems (*Targetsystems*) zu tun haben: Art der Ankopplung der Monitoring-CPU's (*VTA-Targets*), flexible Runtime-Instrumentierung, induzierte Systembeeinflussung, ... .

Aufgrund der (teilweise schon vor Förderungsbeginn geleisteten Vorarbeiten, siehe z.B. [HS1]) befindet sich das Primitive Event Management gegenwärtig im Feinkonzeptions- bzw. im Implementierungsstadium.

**(B) Compound Event Management**

Dieser Bereich ist der Erfassung und Abstraktion (Datenreduktion) der aus den einzelnen Nodes kommenden primitiven Events gewidmet (objektorientierte Spezifikationssprache, distributed Event-Recognition, ...).

Auch hier konnten bereits wesentliche Resultate erzielt werden, vor allem in bezug auf die Spezifikation einer objektorientierten Event Definition Language (GOLD MINE) und den konzeptuellen Entwurf eines distributed Event Recognizers, siehe [St2].

**(C) View Management**

Dieser Bereich beschäftigt sich im wesentlichen mit der "Benutzer-Sicht" des Targetsystems sowie der Analyse/Präsentation von Meßergebnissen, vor allem mit den daraus resultierenden Anforderungen an das funktionale Gesamtkonzept des VTAs.

Hier wurden zunächst grob die momentan bekannten wissenschaftlichen/praktischen Einsatzmöglichkeiten des VTAs analysiert und die daraus resultierenden Anforderungen an das System festgelegt; hierbei stellten sich Flexibilität und leichte Erweiterbarkeit als zentrale Kriterien heraus. Darauf aufbauend wurde ein universelles funktionales Gesamtkonzept (siehe [S2]) bestehend aus einem *Basis-System* und (applikations-spezifischen) *Views* entwickelt, das auf einem speziell dafür entworfenen *Distributed Object Based Operating System* (DOBOS, siehe [SK2]) basiert. Damit ist zunächst ein objektorientiertes Design des VTAs möglich, was nicht zuletzt in Hinblick auf die Entwicklung neuer Views viele Vorteile bringt. Darüberhinaus können in diesem System aber auch existierende "Fremdwerkzeuge" wie etwa Interface-Builder oder spezielle (graphische) Programmiersprachen (für die Realisierung der Datenanalyse und Präsentation) homogen integriert werden.

Daneben gibt es noch einen weiteren Teilbereich, der – über die Bedeutung für den VTA hinausgehend – eine interessante Forschungsarbeit für sich darstellt:

#### (4) **Clock Synchronization**

Im Rahmen des VTAs geht es hier um die Bereitstellung eines clocksynchronisierten pSOS<sup>™</sup> Multiprozessorsystems als Basis für die Implementierung, das in Hinblick auf die zeitliche Ordnung von Events unbedingt benötigt wird.

Im originalen Projektantrag wurde (aus pragmatischen Gründen) eine softwaremäßige Lösung des Problems angepeilt, die jedoch mit einer relativ geringen Genauigkeit der Zeitmessungen des VTAs bezahlt werden muß. In einem Zusatz zum originalen Projektantrag (*Kooperationen mit anderen Forschungsstätten*) wurde auf die Zweckmäßigkeit bzw. Möglichkeit einer Hardware-Lösung auf Basis der Kombination von *Deterministic Ethernet* (INRIA) und der *Clock Synchronization Unit* (TU Wien) verwiesen, allerdings wurden die hierfür notwendigen Zusatzmittel damals nicht explizit beantragt und daher auch nicht bewilligt.

Angesichts der Wichtigkeit dieser Thematik wurden im Zuge der Projektarbeit dennoch Aktivitäten in Richtung der erwähnten Hardware-Lösung gesetzt. Konkret wurde die Spezifikation eines Hardware-Moduls *Network Timestamp Unit* (NTSU) erarbeitet, welches den originalen Ethernet-Controller der als VTA-Targets eingesetzten Force CPU-30 Boards (der als Piggyback realisiert ist) ersetzen soll.

Alle diese Punkte wurden im wesentlichen auch schon im *Projektantrag* (Seite 1-3 bis 1-5) dargelegt (der Bereich (D) wurde ursprünglich als Teil von (A) geführt). Auch die damalige Projektorganisation (Seite 2-8, 2-9) trug dieser Unterteilung Rechnung.

Im Zuge der tatsächlichen Arbeit am Projekt stellte sich heraus, daß eine Abänderung der ursprünglichen Projektorganisation angebracht war. Wie im *Zwischenbericht für das Jahr 1991* näher ausgeführt wurde, erschien es günstig, den Bereich (D) aus (A) herauszulösen und dessen Bearbeitung etwas aufzuschieben. Diese Entscheidung zusammen mit einer kürzlich eröffneten Kooperationsmöglichkeit mit einem Teilnehmer am ESPRIT-Projekt *Eurochip* eröffnet nun die Möglichkeit, eine (für uns normalerweise kaum finanzierbare) Erweiterung der Projektziele in Punkto erzielbare Meßgenauigkeit und – zumindest mittelfristig – induzierte Systembeeinflussung ins Auge zu fassen; letzteres wurde schon von den Fachgutachtern im Zusammenhang mit dem originalen Antrag als sehr wichtig erachtet. Allerdings macht dies zusätzliche Förderungsmittel erforderlich (siehe Abschnitt 2.2).

Darüberhinaus ergibt sich infolge der resultierenden Verringerung der Komplexität des Bereiches (A) die Möglichkeit, die ebenfalls schon von den Fachgutachtern als notwendig erachteten, aber aus pragmatischen Gründen hintangestellten Erweiterungen hinsichtlich der *Erfassung externer Ereignisse (Interrupts)* zu realisieren, wobei aber ebenfalls zusätzliche Förderungsmittel unabdingbar sind (siehe Abschnitt 2.1).

Nicht zuletzt wurde im Zuge der Arbeit an dem Projekt evident, daß der Umfang und die Komplexität des Bereiches (C) neben einer Ausweitung des Personalstandes auch zusätzliche Mittel erforderlich macht. Hier muß ohne Umschweife eingestanden werden, daß der originale Projektantrag nicht weitreichend genug war. Auf der anderen Seite erlauben es die mittlerweile gewonnenen Erfahrungen bzw. Resultate, einen wohldefinierten Zusatzbedarf zu benennen, der insgesamt ebenfalls zu einer wesentlichen Verbesserung der erwarteten Projektergebnisse führt (siehe dazu Abschnitte 2.3 und 2.4).

## 2. Beantragte zusätzliche Förderungsmittel

### 2.1 Erfassung externer Ereignisse

Wie schon im damaligen *Projektantrag* (unter Erweiterungsmöglichkeiten, Seite 2-5) erwähnt – und von den Fachgutachtern bestätigt – wäre es nötig, auch Ereignisse erfassen zu können, die eigentlich auf elektrischem (Hardware-)Level vorliegen. Auf der Basis von Software-Events allein ist es nämlich nicht möglich, irgendwelche Zeitverzögerungen durch die Prozeßperipherie zu berücksichtigen. Dieser Umstand betrifft etwa Interrupt-auslösende Eingänge, über die üblicherweise das Vorliegen von Sensordaten gemeldet werden.

Die Bereitstellung von Möglichkeiten zur Erfassung externer Ereignisse im Rahmen des Projektes VTA hätte konkret folgende Vorteile:

- *Genaue Erfassung von end-to-end Reaktionszeiten* (z.B. von Sensor-Interrupt bis zu Aktor-Stellsignal) wird möglich.
- *Einsatz des VTAs für "ereignisorientierte" Meßprobleme\**, die mit Abtast-Verfahren (z.B. Logikanalysatoren) nicht (oder nur schwer) zu lösen sind: Über lange Meßperioden kontinuierliche, hochgenaue Erfassung und Auswertung mehrerer Ereignisse, wobei zwischen dem Eintreten ein und desselben Ereignisses relativ große Zeitintervalle liegen (Sporadic Events). Ein Logikanalysator müßte hierfür die Event-signalisierenden Leitungen kontinuierlich (mit der geforderten hochgenauen Auflösung!) abtasten und parallel dazu alle Triggerbedingungen testen (und natürlich die Langzeit-Datenauswertung durchführen), während in Wirklichkeit nach dem Eintreten eines bestimmten Ereignisses für relativ lange Zeit kein weiteres solches Ereignis kommen kann.

Ein konkretes Problem aus dem Umfeld des Projektes VTA ist die kontinuierliche Erfassung der Synchronisationsgenauigkeit mehrerer Computer-Uhren (NTSUs), die (für Meßzwecke) ein digitales 1 PPS-Signal liefern (d.h., periodisch "ihren" Sekundenbeginn durch eine positive Flanke auf einer digitalen Leitung signalisieren). Gesucht ist unter anderem etwa ein Histogramm der Zeit zwischen der positiven Flanke der jeweils "frühesten" und "spätesten" Uhr, wobei die Auflösung etwa 100 ns sein sollte.

Die Realisierung der Erfassung externer Ereignisse erfordert zunächst einmal einige *zusätzliche Hardware-Komponenten* innerhalb des VTAs, ohne die externe Ereignisse überhaupt nicht zugänglich sind. Über diese Zusatz-Hardware kann der VTA Zustands- oder Wertänderungen der Signale erfassen und als spezielle externe Ereignisse verarbeiten. Rein technisch gesehen ist es hierzu notwendig, Peripheriekomponenten für die verschiedensten elektrischen Signalformen (digitale und

---

\* Dieser Punkt bekommt in Hinblick auf die Ausführungen des Abschnittes 2.2.1 zusätzliches Gewicht.

analoge Signale) vorzusehen, an die die interessierenden Leitungen anzuschließen sind.

Die benötigte Flexibilität hinsichtlich der verschiedensten Signale kann nun relativ einfach und kostengünstig auf Basis eines der am Markt existierenden modularen *I/O-Systeme* (bestehend aus den verschiedensten I/O-Modulen wie Digital I/O, Analog I/O, Timer/Counter, ... auf VME-Trägerplatinen) realisiert werden. Besonders geeignet erscheint hier das von der Firma MEN initiierte, offene Konzept der M-Module. Diese Präferenz ist die Folge eines Vergleiches von ähnlichen Systemen (u.a. Oettle+Reichler, Force, PEP), wobei als Hauptkriterium ein möglichst guter Kompromiß zwischen Leistungsfähigkeit des Systems (z.B. 16-bit Bus) auf der einen und einfacher Anschaltung (eigenentwickelter) Module auf der anderen Seite diene.

Konkret ist folgende Minimalkonfiguration erforderlich:

| Nr. | # | Bezeichnung | Bemerkungen                                    | öS (incl. 20%) |
|-----|---|-------------|------------------------------------------------|----------------|
| 1   | 1 | CPU-30 ZBE  | CPU-Modul für External Event Preprocessing     | 53.964,--      |
| 2   | 2 | MEN A201    | Trägerplatine für M-Module, a DM 760,--        | 12.768,--      |
| 3   | 2 | MEN M1      | M-Mod Digital Input (Sink), a DM 350,--        | 5.880,--       |
| 4   | 2 | MEN M21     | M-Mod Digital Input (Source), a DM 380,--      | 6.384,--       |
| 5   | 1 | MEN M5      | M-Mod Analog Input, a DM 1.260,--              | 10.584,--      |
| 6   | 2 | MEN M13     | M-Mod Timer/Counter, a DM 770,--               | 12.936,--      |
| 7   | 2 | MEN M0      | M-Mod für Prototypenentwicklung, a DM 125,--   | 2.100,--       |
| 8   | 2 | MEN M3      | M-Mod Digital Output (Testphase), a DM 370,--  | 6.216,--       |
| 9   | 1 | MEN M4      | M-Mod Analog Output (Testphase), a DM 1.200,-- | 10.080,--      |
| 10  | 1 | MENCOMP2    | 5 slot VME_Rack inkl. Netzteile, a DM 3.300,-- | 27.720,--      |

Die vorgestellte Konfiguration ist insofern flexibel, als sie sowohl den Erfordernissen des VTAs selbst genügt als auch die für die Testphase notwendigen Möglichkeiten bietet. Abschließend sollte erwähnt werden, daß es zu dem (relativ teuren) Force CPU-30 Modul insofern keine Alternative gibt, als die im vorigen Abschnitt, Punkt (D) erwähnte NTSU für dieses Modul entwickelt wird.

Die Erfassung externer Events bedingt natürlich auch eine Berücksichtigung innerhalb des (Software-)Konzeptes des VTAs, und zwar hauptsächlich innerhalb des Primitive Event Managements. Dies ist insofern nicht trivial, als die Vielfalt der möglichen Prozeßsignale äußerste Flexibilität erfordert: Neben einfachen Dingen wie der Zustandsänderung eines digitalen Signales sind zum Beispiel auch



Wertänderungen analoger Größen durchaus als Trigger denkbar; die PEM-Software muß daher (leicht erweiterbare!) Möglichkeiten für die *Spezifikation\** derartiger Triggerbedingungen bieten.

Daraus resultiert einerseits eine beträchtliche Erweiterung der (konzeptuellen) Komplexität des Primitive Event Managements und andererseits zusätzlicher Implementierungsaufwand hinsichtlich der externen Eventerfassung und der notwendigen Testumgebung. Für die Implementierungsarbeit wird ein zusätzliches Förderungstipendium für einen Diplomanden benötigt:

| Nr. | # | Bezeichnung | Bemerkungen                                                                | öS (incl. 20%) |
|-----|---|-------------|----------------------------------------------------------------------------|----------------|
| 11  | 1 |             | Förderungstipendium (1 Jahr, öS 5.000,--/Monat)<br>Externe Event-Erfassung | 60.000,--      |

Abschließend ist festzuhalten, daß auch von den Fachgutachtern des FWF auf die Notwendigkeit der vorgestellten Erweiterung hingewiesen wurde. Allerdings erschien es zum Zeitpunkt des Projektantrages nicht sinnvoll, dies innerhalb des Forschungsprojektes zu realisieren. Die Gründe dafür waren insofern pragmatischer Natur, als das Gesamtprojekt (und die Förderung) nicht durch die Berücksichtigung zu vieler Teilaspekte gefährdet werden sollte.

## 2.2 Clock-Synchronisation

Eine zuverlässige globale Uhrzeit ist wohl für jedes verteilte Echtzeitsystem von entscheidender Bedeutung. Obwohl es sich wissenschaftlich um beileibe kein "jungfräuliches" Gebiet handelt, sind die Grenzen der Thematik noch lange nicht erreicht. Dies schlägt sich unter anderem auch darin nieder, daß immer wieder zahlreiche Publikationen zu diesem Thema (siehe z.B. [Cri89], [Arv89], [Ram90], [Ran91], [Cou91], [Ol91], [Pf91], [Ver91], [Mav92] für einige neuere Arbeiten) erscheinen. Auch die Akzeptanz in der Industrie ist bis jetzt – trotz des Bedarfes – sehr gering.

Einige Punkte, an denen noch einiger Forschungsaufwand zu investieren wäre, sind

- Verbesserung der erreichbaren Synchronisationsgenauigkeit, v.a. durch weitergehenden Hardware-Support (siehe z.B. [Kop87], [Ram90]) und den Einsatz neuer Algorithmen (u.a. [Ran91], [Cou91], [Pf91]).
- Verbesserte Integration von Clocksynchronisations-Verfahren mit hochgenauen Master-Clocks.
- Experimentelle (Langzeit-)Untersuchungen der verschiedenen Verfahren.

---

\* Natürlich ist die Erkennung des Eintretens einer Triggerbedingung selbst kein schwieriges Problem.

- Über worst/average-case hinausgehende theoretische Analyse der verschiedenen Verfahren.

Wie bereits zum Zeitpunkt des Projektantrages, im *Zusatz zum Projektantrag (Kooperationen mit anderen Forschungsstätten)* erläutert, wäre es aus Gründen der erzielbaren Meßgenauigkeit (zeitlichen Auflösung) günstig, auch bei der Realisierung des VTAs eine hardwaremäßige Lösung des Clocksynchronisations-Problems vorzusehen; diese sollte auf Basis des von INRIA Rocquencourt entwickelten *Deterministic Ethernet* und des *Clock Synchronization Unit (CSU)* Chips der TU-Wien entwickelt werden. Es wurde schon damals auf die Notwendigkeit zusätzlicher Mittel hierfür hingewiesen (die jedoch nicht explizit beantragt und daher auch nicht bewilligt wurden).

Die konkreten Vorteile der Verwendung einer Hardware-Clocksynchronisation im VTA wären:

(1) *Wesentliche Erhöhung der Meßgenauigkeit des VTAs*

Die Meßgenauigkeit des VTAs wird zunächst einmal von (1) dem indeterministischen Anteil der Latenzzeit zwischen dem Eintreten eines Events im Targetsystem und dem darauffolgenden Auslesen der lokalen Uhr (= Time-stamping) im VTA-Target sowie (2) der Synchronisationsgenauigkeit der lokalen Uhren untereinander bestimmt. Beide Einflußgrößen können durch die Hardware-Clocksynchronisation wesentlich verringert werden:

- Nach ersten Abschätzungen sollte sich mit dem projektierten Ansatz eine Synchronisationsgenauigkeit der lokalen Uhren untereinander in der Gegend von 1–2  $\mu\text{s}$  erreichen lassen; gegenüber der ursprünglich vorgesehenen Genauigkeit von 20–50  $\mu\text{s}$  bedeutet dies eine Verbesserung um mehr als eine Größenordnung.
- Die Hardware-Lösung eröffnet die Möglichkeit, das Event-Timestamping (teilweise) ebenfalls in Hardware zu realisieren; ein derartiger Ansatz wird in anderen (Hardware/Hybrid-)Monitoring-Systemen wie z.B. ZM4 ([Hof92]) oder TOPSYS ([Bem90]) verwendet. Die so erreichbaren Latenzzeiten im Bereich von 100 ns bedeuten gegenüber der ursprünglichen, auf Basis eines hochpriorisierten Interrupts geplanten Software-Lösung (mit Latenzzeiten im Bereich von 5–10  $\mu\text{s}$ ) ebenfalls eine wesentliche Verbesserung.

Zusätzliche Auswirkung auf die Meßgenauigkeit (im Falle von Software-Events) hat natürlich (3) der durch den Instrumentierungs-Code verursachte Overhead. Während das ursprüngliche Konzept mit einem Overhead in der Gegend von 5–50  $\mu\text{s}$  rechnete, haben unsere Ergebnisse bezüglich des Primitive Event Managements (siehe [HS1]) gezeigt, daß 0.5–5  $\mu\text{s}$  realistisch zu erreichen sind.

Alle diese Argumente zeigen, daß eine Verbesserung der Meßgenauigkeit von 50–100  $\mu\text{s}$  auf 5–10  $\mu\text{s}$ , also um eine Größenordnung, realisierbar ist. Daß eine solche Verbesserung auch notwendig ist, resultiert aus den

stets steigenden Verarbeitungsleistungen der Targetsystem-Prozessoren, die selbst innerhalb der nun erreichbaren 5-10  $\mu$ s Timestamp-Ungenauigkeiten schon dutzende Befehle exekutieren können!

## (2) *Wesentliche Steigerung der Verarbeitungsleistung des VTAs*

Eine sehr wichtige Kenngröße eines Monitoring-Systems ist natürlich die Anzahl der verarbeitbaren Events/Sekunde. Wenngleich auch hier insofern keine Echtzeitbedingungen vorliegen, als das Monitoring-System bei der Auswertung ohne weiteres hinter dem Targetsystem "nachhinken" kann, so setzen doch letztlich die endlichen Ressourcen (v.a. Pufferkapazitäten) der Verarbeitungsleistung eine bestimmte Grenze.

Da es sich bei dem VTA um ein verteiltes Monitoring-System handelt, kommt der Übertragungskapazität des Kopplungsmediums (VTA-Ethernet) wesentliche Bedeutung zu. Der ursprüngliche Ansatz zur Clocksynchronisation hätte nun ein TDMA-Schema auf Standard-Ethernet vorgesehen (wie in MARS, siehe [Kop89]), was die Nutzkapazität drastisch verringern würde.

Der hier vorgeschlagene Ansatz mittels Deterministic Ethernet vermeidet die übliche (indeterministische) CSMA/CD-Kollisionsbehandlung und erlaubt es daher, die Hardware-Clocksynchronisation auch ohne TDMA zu realisieren. Damit steht dem VTA die volle Kapazität des Ethernets zur Verfügung.

### 2.2.1 Lösungsalternativen

An dieser Stelle erscheint es zunächst angebracht, auf die von einem der Fachgutachter erwähnte Alternative zur Clock-Synchronisation einzugehen. Die ursprüngliche Anregung war, statt synchronisierter Clocks einfach mehrere DCF-77 Funkuhren einzusetzen. Diese Lösung hätte jedoch aufgrund der – technisch bedingten – mangelnden Synchronisationsgenauigkeit der gegenwärtig angebotenen Uhren untereinander (z.B. Meinberg VME-Funkuhr: 10-50  $\mu$ s) erst recht einen Synchronisations/Korrekturmechanismus (oder eventuell ein diffiziles PLL-Empfängerdesign?) erfordert, um auf die angestrebte Genauigkeit (in der Gegend von 1-2  $\mu$ s) zu kommen.

Ein bezüglich der erreichbaren Synchronisationsgenauigkeit wesentlich besseres System wäre das GPS (Global Position System), das auf der Frequenz von ca. 1.5 GHz arbeitet und tatsächlich die Entwicklung hochgenauer Funkuhren zuläßt. Die Probleme mit diesem System in Verbindung mit einem (nicht-stationären, also transportablen!) System wie dem VTA sind jedoch unter anderem folgende:

- Ein GPS-Empfänger benötigt auf jeden Fall eine Dachantenne.
- Die für das GPS praktikable Master/Slave-Architektur\* (die de facto auch eine Form von Clocksynchronisation einsetzt!) benötigt zusätzlich zum

---

\* Ein GPS-Empfänger dient als Master, der an die angeschlossenen Slaves seine Zeitinformation (z.B. über Zeitcodes wie IRIG x) überträgt.

VTA-Ethernet eine separate Verbindung (Zeitkanal) der VTA-Targets untereinander.

- Bei Einsatz existierender GPS VME-Funkuhren (wie sie z.B. von der Fa. Lange/Olching vertrieben wird) ist in jedem Node des Targetsystems ein weiterer Slot für die GPS-Uhr (unter Heranziehung des Target-VME-Busses zur Kommunikation!) notwendig.

Somit ist festzuhalten, daß eine Lösung der Zeitproblematik auf der Basis terrestrischer oder satellitengestützter Funkuhren für den VTA der Realisierung einer Clocksynchronisations-Hardware in einigen wichtigen Punkten unterlegen sein dürfte. Es ist jedoch – wie schon eingangs erwähnt – sicherlich sehr wichtig und lohnend, die Möglichkeiten/Grenzen der vorgestellten Systeme systematisch zu erforschen und zu vergleichen. Dies sollte am besten im Rahmen eines eigenen Forschungsprojektes erfolgen, welches unter anderem folgende Ziele haben könnte:

- (1) Experimentelle Erforschung der Möglichkeiten und Grenzen (Genauigkeit, Fehlertoleranzaspekte, ...) von Systemen zur Bereitstellung einer globalen Zeit in verteilten Rechnersystemen.
- (2) Versuch einer organischen Integration von Clocksynchronisations-Verfahren und hochgenauen Master-Clocks, die die Vorteile beider Verfahren optimal kombiniert.

Es erscheint in diesem Zusammenhang wichtig, darauf zu verweisen, daß ein derartiges Projekt auf folgende Ergebnisse des Projektes VTA zurückgreifen könnte:

- Die projektierte Clocksynchronisations-Hardware, die (natürlich) entsprechend universell gestaltet wird.
- Die im vorigen Abschnitt erwähnte Erfassung und Auswertung externer Ereignisse mit Hilfe des VTAs, der somit (unter anderem) das optimale System zu einer entsprechenden experimentellen Performance-Analyse darstellt.

### 2.2.2 Realisierung der NTSU

Konkret gelang es in der Zwischenzeit – durch die bereitwillige Unterstützung von G. LeLann/INRIA Roquencourt und Hermann Kopetz/TU Wien –, ein relativ detailliertes Konzept für eine dem ursprünglichen Konzept gemäße *Network Timestamp Unit* (NTSU) zu entwickeln. Dabei stellten sich einige Schwierigkeiten heraus:

- (1) Bei der Realisierung der originalen CSU im Jahre 1985/1986 mußten – aufgrund der zum Entwicklungszeitpunkt verfügbaren Technologie (2 µm CMOS Standardzellen) – relativ große Beschränkungen der Funktionalität

in Kauf genommen werden, um die Chipkomplexität gering (4000 Gatter-äquivalente) zu halten (siehe beiliegendes Datenblatt der CSU).

(2) Der Einsatz der CSU in unserer NTSU macht daher die externe Realisierung fehlender bzw. zusätzlicher Funktionalität erforderlich, unter anderem

- "Verlängerung" des internen 20-bit Counters (Auflösung 1  $\mu$ s) mit einer Möglichkeit zum (Interrupt- und DMA-)atomaren Auslesen des gesamten Zählerstandes. Dies stellt sich als eine extern sehr komplexe Erweiterung dar, die intern praktisch "kostenlos" zu erreichen wäre.
- Erhöhung der Auflösung auf 100 ns.
- Erweiterung der Datenbusbreite sowie aller weiteren internen Register auf 32 Bit, um auch hier die Probleme mit dem Interrupt- und DMA-atomaren Auslesen der diversen CSU-Register zu beheben; die externe Realisierung erfordert einen komplexen Bus-Arbiter, der (kurze) unvorhersehbare Delays induziert (was für die Genauigkeit der Clocksynchronisation schlecht ist).
- Erweiterten Timestamping-Support für Events. Die Hauptaufgabe der NTSU im direkten Zusammenhang mit dem VTA ist es ja, ankommende Events mit einem Zeitstempel zu versehen. Die im originalen Antrag angepeilte Methode war es, dies mit Hilfe einer Interrupt Service Routine zu realisieren. Diese pragmatische Vorgangsweise hat jedoch – wie schon erwähnt – den Nachteil, daß durch die indeterministische Latenzzeit relativ große Meßungenauigkeiten verursacht werden.  
Die wesentlich bessere Methode wäre es, die CSU mit einem (zusätzlichen) Hardware-Zeitstempelmechanismus zu versehen, der mit einem (vorzugsweise externen) FIFO kombinierbar ist.
- Diverse Erweiterungen wie zusätzliche Timeout- und Interrupt-Features und vor allem die Herausführung aller jener Signale, die den externen Ausbau der CSU (etwa bzgl. Timestamp-Länge, FIFO, ...) erlauben; dies ist bei der vorhandenen CSU unmöglich.

(3) Die mit der vorhandenen CSU mögliche Realisierung der NTSU erfordert ca. 30 relativ große Chips (neben den Controllern und RAMs allein 6 oder 7 GALs für die externe Logik), was einige praktische Probleme in Hinblick auf Stromverbrauch, Timing (25 MHz) und vor allem auch Platzbedarf bedingt; hierbei ist zu beachten, daß das zu ersetzende Ethernet-Piggyback der Force CPU-30 nur 3.5 x 8.5 cm groß ist und 14 Chips trägt.

Angesichts dieser Probleme ergab es sich von selbst, die Möglichkeiten zur Entwicklung eines neuen CSU ASICs zu sondieren und auf diese Weise die Komplexität der NTSU selbst zu verringern. Die Vorteile dieser Vorgangsweise – abgesehen von den Vorteilen für das im vorigen Abschnitt erwähnte, zusätzliche

Forschungsprojekt – wären:

- *Vereinfachung der Entwicklung der NTSU selbst*

Die Werkzeuge für die ASIC-Entwicklung erlauben die Simulation des Bausteines, wodurch – im Gegensatz zur eigentlichen NTSU – viele Designfehler noch vor der Realisierung gefunden werden können. Daneben verringert sich die Komplexität der NTSU wesentlich.

- *Möglichkeit der Entwicklung einer speziellen Monitoring-Hardware*

Das der Prototyp-Realisierung des VTAs zugrundeliegende Konzept der Erfassung der aus dem Targetsystem kommenden Events verwendet ein Standard-Feature der Force CPU-30 (Force Message Broadcast, FMB) in Verbindung mit shared Memory zur Signalisierung/Datenübertragung. Dies verursacht natürlich eine Beeinflussung des Targetsystems, die durch ein stärker hardwareunterstütztes Monitoring geringer gehalten werden könnte, siehe z.B. [Bem90].

Durch ein entsprechendes Design der neuen CSU wäre es möglich, auf die hierfür notwendigen Erfordernisse Rücksicht zu nehmen, sodaß die spätere Entwicklung eines echten Hardware-Monitorings wesentlich vereinfacht würde.

Um die Realisierbarkeit all dieser Ideen zu verifizieren, erfolgte eine Kontaktaufnahme mit (Elektrotechnik-)Instituten der TU-Wien, die zu folgenden Ergebnissen führte:

- **Institut für Technische Informatik, Abt. Softwaretechnologie (Prof. Kopetz)**

An der genannten Abteilung wurde die originale CSU (im Rahmen des MARS-Projektes) entwickelt. Prof. Kopetz erklärte sich bereit, die Entwicklung einer neuen CSU zu unterstützen.

- **Institut für Elektrische Meßtechnik, Abt. für Meßsysteme (Prof. Schweinzer)**

An der genannten Abteilung läuft gegenwärtig ein FWF-Projekt P7582-TEC (*Sichere Computersysteme*), welches die Entwicklung einer M88000-Hardware für das MARS-System (Prof. Kopetz) zum Ziel hat. Da das MARS-System auf der Verwendung der CSU aufbaut, ergibt sich die glückliche Konstellation, daß auch in der Folge dieses Projektes die Entwicklung eines Boards erforderlich sein wird, welches unserer NTSU sehr ähnlich ist.

Demzufolge ergibt sich die Möglichkeit, den Synergieeffekt dieser parallelen Bedürfnisse auszunutzen. Prof. Schweinzer bzw. Univ. Ass. Andreas Steininger haben sich bereiterklärt, uns bei der Konzeption und Entwicklung der NTSU zu unterstützen.

- **Institut für Computertechnik (Prof. Dietrich)**

Die genannte Abteilung ist Teilnehmer am ESPRIT-Projekt *Eurochip* und hat auf diese Weise die Möglichkeit, ASICs zu entwickeln und bei den beteiligten Halbleiterherstellern fertigen zu lassen. Die dafür aufzuwendenden Kosten betragen einen Bruchteil der hierfür normalerweise notwendigen

Mittel.

Prof. Dietrich bzw. Univ. Ass. Dietmar Loy haben sich bereiterklärt, im Rahmen der Entwicklung eines neuen CSU-ASICs für die NTSU mit uns zu kooperieren.

Natürlich erfordert dies aber auch die Bereitstellung zusätzlicher Förderungsmittel. Im Detail sind dies zunächst ein *vollbeschäftigter Vertragsassistent/1 Jahr* (Elektrotechnik), der die Entwicklung des CSU ASICs und der NTSU durchführen soll. Die Stelle sollte am Institut für Computertechnik angesiedelt werden, wodurch die Infrastruktur/Betreuung des Eurochip-Projektes für die Entwicklung des CSU ASICs verfügbar ist. Für diese Aufgabe ist weiters ein *Einschulungskurs* in die Entwicklungs-Werkzeuge (bei Firma ES2/München) sowie ein Satz *Dokumentation* notwendig.

Darüberhinaus erschien es zweckmäßig, sich der beratenden Tätigkeit des ursprünglichen Entwicklers der CSU, **Dr. Wilhelm Ochsenreiter**, zu versichern. Dieser ist mittlerweile (voll) in der Privatwirtschaft tätig und sollte mit einem *Konsulentenhonorar* für seine Mitarbeit entschädigt werden; auf diese Weise kann der Designprozeß sicherlich wesentlich rascher und effektiver abgewickelt werden.

Ebenfalls notwendig sind natürlich auch die finanziellen Mittel für die *Fertigung einer Kleinserie* (20 Stück) des CSU ASICs. In den dafür zu kalkulierenden Mittel sind zu berücksichtigen:

- Chipkomplexität (→ Chipfläche), wobei die CSU sicherlich ein recht komplexer Baustein wird (Größenordnung 10000-15000 Gatteräquivalente). Die im Rahmen von ESPRIT Eurochip am ehesten in Frage kommende 1.5 µm CMOS-Technologie läßt ca. 300 Gatter/mm<sup>2</sup> Chipfläche zu, wobei die Kosten 110 ECU/mm<sup>2</sup> (excl. Mwst.) betragen.
- Gehäuse (Anzahl der Pins), wobei aufgrund der erwünschten externen Erweiterbarkeit ("Kaskadierbarkeit") ca. 100 Pins notwendig werden; hierfür muß eine zusätzliche Chipfläche von 12 mm<sup>2</sup> für das Bonding eingerechnet werden.
- 1 komplettes Redesign, das bei so komplexen Bausteinen erfahrungsgemäß immer eingerechnet werden muß.

Für die Entwicklung der NTSU selbst werden folgende finanzielle Mittel benötigt:

- Entflechtung und Herstellung eines Multilayer-Boards (10 Stück) sowie Bestückungskosten (SMD-Technik!); auch hier muß ein vollständiges Redesign eingerechnet werden.
- Bauteilkosten für 7 NTSU-Boards (1 Prototyp + 6 für die vorhandenen Force CPU-30 Boards), die sich auf ca. öS 4.000,--/Board belaufen.

Ebenfalls notwendig ist schließlich ein weiteres Force CPU-30-Board inklusive eines VME-Racks, ohne das die Entwicklung bzw. der Test der NTSU am Institut

für Computertechnik nicht möglich ist. Hierzu ist zu beachten, daß die NTSU als Piggy-Back auf der Force CPU-30 aufzusetzen ist, was einerseits eine Anpassung an die (leider nicht dokumentierte) Schnittstelle und andererseits einige Änderungen der auf der CPU-30 befindlichen Logik (GALs) notwendig macht.

Die folgende Tabelle faßt die notwendigen Mittel zusammen:

| Nr. | # | Bezeichnung | Bemerkungen                                    | öS (incl. 20%) |
|-----|---|-------------|------------------------------------------------|----------------|
| 12  | 1 |             | Vollbeschäftigter Vertragsassistent            | 440.000,--     |
| 13  | 1 |             | Einschulungskurs ASIC-Tools (Fa. ES2, München) | 30.000,--      |
| 14  | 1 |             | Satz Dokumentation für ASIC-Tools              | 20.000,--      |
| 15  | 1 |             | Kosten für die Fertigung des neuen CSU ASICS   | 240.000,--     |
| 16  | 1 |             | Konsulentenhonorar Dr. W. Ochsenreiter         | 20.000,--      |
| 17  | 1 |             | Kosten für Entwicklung/Fertigung der NTSUs     | 70.000,--      |
| 18  | 1 | CPU-30 ZBE  | CPU-Modul als Carrier für NTSU Piggyback       | 53.964,--      |
| 19  | 1 | MENCOMP2    | 5 slot VME-Rack inkl. Netzteile, a DM 3.300,-- | 27.720,--      |

Abschließend ist noch zu erwähnen, daß die vorgeschlagene Erweiterung der Projektziele auch insoferne positive "Seiteneffekte" zeigt, als sie eine verstärkte interuniversitäre Kooperation mit sich bringt – eine angesichts der leider oft anzutreffenden "Isolation der Wirkungsbereiche" strukturell sehr wichtige Tatsache.

### 2.3 View Management

Wie bereits im Abschnitt 1 erwähnt wurde im Zuge der Arbeit am View Management (Bereich (C)) relativ bald evident, daß dessen Detail-Komplexität und -Umfang im originalen Projektantrag unterschätzt worden war. Wie zum Teil im *Zwischenbericht für das Jahr 1991* näher ausgeführt wurde diesem Problem von unserer Seite aus folgendermaßen begegnet:

- Ein weiterer Univ. Ass. der Forschungsstelle (*DI Harald Haberstroh*) konnte zur (vollen) Mitarbeit am Projekt gewonnen werden.
- Der bewilligte 2. Dienstvertrag (*DI Wolfgang Kastner*) wurde vom Bereich (B) abgezogen und dem Bereich (C) zugeordnet.

Dennoch erfordert die erfolgreiche Realisierung des mittlerweile erstellten funktionalen Gesamtkonzeptes zusätzliche Förderungsmittel, die im folgenden detailliert werden. Wenngleich natürlich nicht bestritten werden soll, daß diese Mittel schon im ursprünglichen Antrag hätten beantragt werden müssen, so erlaubt es die hiermit erfolgende "Nachreichung", einen wohlbegründeten (und damit kleineren)



Bedarf zu benennen, der noch dazu insofern eine Verbesserung der Projektergebnisse erlaubt, als diese durch die schon geleistete Arbeit konkretisiert werden konnten.

### 2.3.1 Distributed Object-Based Operating System DOBOS

Für die Realisierung des VTAs wurde nach einem (zunächst pragmatischen) Ansatz gesucht, wie auf einer verteilten, heterogenen Architektur bestehend aus einer Sun-Workstation (SunOS UNIX) und mehreren 68030-CPU's (pSOS<sup>+</sup>), die durch ein TCP/IP-Netzwerk verbunden sind, ein homogenes Betriebssystem (inklusive Implementierungssprache) mit folgenden Eigenschaften bereitgestellt werden kann:

- Objektorientierung der Applikationssoftware (Inheritance, Polymorphismus, usw.), jedoch ohne die Verwendung einer speziellen objektorientierten Programmiersprache,
- Static Type-Checking,
- (Object) Location Transparency,
- parallele Prozesse,
- objektorientierte Datenbank (konkret persistente Objekte),
- möglichst einfache Anbindung existierender (C-)Software,
- minimaler System-Overhead, einfache Abbildung auf SunOS, pSOS<sup>+</sup> und ANSI C.

Das Resultat der bisherigen Bemühungen ist das Detailkonzept eines dem SunOS und pSOS<sup>+</sup> überlagerten Betriebssystems DOBOS, welches eventuell auch ein Ausgangspunkt für eine eigene Forschungsaktivität auf dem Gebiet der objektorientierten/parallelen Systeme werden könnte, siehe [SK2].

Für die Implementierung von DOBOS wird nun ein weiteres *Förderungsstipendium* für einen Diplomanden benötigt:

| Nr. | # | Bezeichnung | Bemerkungen                                                      | öS (incl. 20%) |
|-----|---|-------------|------------------------------------------------------------------|----------------|
| 20  | 1 |             | Förderungsstipendium (öS 5000,--/Monat)<br>Implementierung DOBOS | 60.000,--      |

### 2.3.2 Views

Wie in [S2] näher erläutert basiert das Design des VTAs auf einem Basis-System in Verbindung mit beliebigen, auf bestimmte Anwendungsgebiete zugeschnittenen Views.

Bei der Realisierung des Basis-Systems ist (durch die "Verschiebung" des bewilligten DVs von Bereich (B) nach (C)) die Implementierung der von Univ. Ass. DI Stefan Stöckler konzipierten Distributed Event Recognition noch offen. Für diese Tätigkeit, die für das Gesamtprojekt von zentraler Bedeutung ist, sollte daher ein zusätzliches *Förderungsstipendium* vorgesehen werden.

In Punkto Realisierung der Views selbst ist für den Prototypen des VTAs die Implementierung zweier einfacher Views vorgesehen, und zwar

- *System View*

Der System View erlaubt es, das Targetsystem auf der Maschinencode-Ebene zu betrachten; so können z.B. Events auf bestimmte Adressen gesetzt werden.

Konzeptuell ist allerdings auch der System View insofern universell gestaltet, als er mit Hilfe eines *Processor(-Board) Support- und Operating System Support-Packages* an die vom Targetsystem vorgegebenen Erfordernisse angepaßt werden kann.

- *C High Level Language View*

Der High Level Language View erlaubt es, das Targetsystem auf der Ebene von Source Code zu betrachten; hier können Events auf bestimmte Sourcecode- Statements gesetzt werden.

Konkret soll der zu implementierende View den Microtec C-Compiler unterstützen, einige der diesbezüglichen konzeptuellen Vorarbeiten wurden bereits erledigt (siehe [H1]).

Für diese Implementierungstätigkeiten werden noch 2 zusätzliche *Förderungsstipendien* benötigt, sodaß sich insgesamt folgender Bedarf ergibt:

| Nr. | # | Bezeichnung | Bemerkungen                                                                       | öS (incl. 20%) |
|-----|---|-------------|-----------------------------------------------------------------------------------|----------------|
| 21  | 1 |             | Förderungsstipendium (öS 5.000,--/Monat, 1 Jahr)<br>Distributed Event Recognition | 60.000,--      |
| 22  | 1 |             | Förderungsstipendium (öS 5.000,--/Monat, 1 Jahr)<br>System View                   | 60.000,--      |
| 23  | 1 |             | Förderungsstipendium (öS 5.000,--/Monat, 1 Jahr)<br>High Level Language View      | 60.000,--      |

### 2.3.3 Visualisierung

Im Zuge der Arbeit am View-Management Bereich (C) stellte sich heraus, daß ein Teil der Funktionalität, nämlich die Datenanalyse und Präsentation, ein relativ abgeschlossenes System darstellt. Angesichts der besonderen Bedeutung, die der Datenanalyse und Präsentation in praktisch allen Bereichen der Informatik zukommt, war es natürlich naheliegend, bestehende Visualisierungssysteme hinsichtlich einer möglichen Integration im Gesamtsystem des VTAs zu untersuchen.

Infolge eines glücklichen Zufalles ist zur Zeit eine Projektgruppe am Rechenzentrum der TU-Wien mit einer Evaluation leistungsfähiger Visualisierungssysteme beschäftigt. Tatsächlich existieren nun einige universelle Systeme (vor allem das datenflußorientierte *LabVIEW* und *AVS*), die als sehr brauchbare Kandidaten für eine Integration in den VTA angesehen werden können. Es handelt sich dabei um "graphische" Programmiersprachen bzw. -umgebungen, die keine fix vorgegebene Funktionalität bieten, sondern durch frei programmierbare Module leicht und beliebig zu erweitern sind; dies sind Forderungen, die auch im Zuge unserer Arbeit am View Management als essentiell erkannt wurden.

Angesichts des ungeheuren Entwicklungsaufwandes, der in derartigen Visualisierungssystemen steckt, ist es natürlich klar, daß keine im Rahmen des Projektes angestrebte Eigenentwicklung einen auch nur annähernd vergleichbaren Funktionsumfang bieten kann. Würde nun – quasi als Gerüst – ein derartiges Visualisierungssystem eingesetzt, so könnten die ursprünglich auf konzeptuelle Modelle beschränkten Auswertungswerkzeuge sogar weitgehend innerhalb des Projektes realisiert, also entsprechende Analyse- und Präsentationsmodule entwickelt werden.

Eine von uns durchgeführte, weitere Evaluation von *LabVIEW* und *AVS* fiel zu Gunsten von *LabVIEW* aus. Es sollte daher eine *Lizenz von LabVIEW* für die (aus Projektmitteln stammende, von uns aber intern aufgerüstete) Sun-Workstation angeschafft werden. Zusätzlich wird ein *Förderungsstipendium* für einen Diplomanden benötigt, der (neben der Anbindung von *LabVIEW* an den Rest des VTAs) vor allem die bis jetzt konzipierten Analyse- und Darstellungsfunktionen in der *LabVIEW*-Programmiersprache realisieren soll.

Insgesamt ergibt sich hier folgender zusätzlicher Bedarf an Förderungsmitteln:

| Nr. | # | Bezeichnung | Bemerkungen                                                                  | öS (incl. 20%) |
|-----|---|-------------|------------------------------------------------------------------------------|----------------|
| 24  | 1 |             | Lizenz <i>LabVIEW</i> für Sun SPARCStation                                   | 80.532,--      |
| 25  | 1 |             | Förderungsstipendium (öS 5.000,--/Monat, 1 Jahr)<br>Anbindung <i>LabVIEW</i> | 60.000,--      |

## 2.4 Strukturelle Maßnahmen

Die in den vorigen Abschnitten erläuterten "punktuellen" Förderungsmittel sind allerdings nur dann zielführend, wenn eine *Verlängerung der Projektdauer um 1 Jahr* (Ende 1994) gewährt wird. In diesem Zusammenhang erscheint es angebracht, die angestrebte Projektorganisation zu erläutern.

Zunächst einmal konnte 1992 ein zusätzlicher Univ. Ass. der Forschungsstelle (*DI Harald Haberstroh*) als Leiter für den Bereich des View Managements gewonnen werden. Konkret bedeutet dies, das jetzt insgesamt 3 Univ. Ass. (2 Dissertationen, 1 Habilitation) der Forschungsstelle voll am Projekt mitarbeiten, wobei jedem die Betreuung eines der drei Hauptbereiche obliegt (vgl. Abschnitt 1 sowie den *Zwischenbericht für das Jahr 1991*). Für die Zukunft ist – unter Einrechnung der zusätzlich beantragten Mittel (**fett gedruckt**) – folgende Projektorganisation geplant:

Projektleitung: *Univ. Ass. Dr. U. Schmid*

(1) Primitive Event Management (PEM)

– *Univ. Ass. Dr. U. Schmid*

Leitung

Grobkonzept

– *V.Ass. DI J. Klasek*

Implementierung PEM (Dienstvertrag)

– NN.

**Hard/Software External Event Management (Förderungsstipendium)**

– Praktikanten, Diplomanden

(2) Compound Event Management

– *Univ. Ass. DI S. Stöckler*

Leitung

Design und formales Modell GOLD\_MINE

Konzept Distributed Event Recognition

– NN.

**Implementierung Distributed Event Recognition (Förderungsstipendium)**

– Praktikanten, Diplomanden

Implementierung GOLD\_MINE Toolset

(3) View Management

– *Univ. Ass. DI H. Haberstroh*

Leitung

Konzept/Modellierung Human IFC

Konzept/Modellierung Views

– *V.Ass. DI W. Kastner*

Feinkonzeption DOBOS (restl. Dienstvertrag)

– NN.

**Implementierung DOBOS (Förderungsstipendium)**

– *J. Divisch*

- Feinkonzept/Implementierung Human IFC VTA-Basissystem  
(Förderungsstipendium)
- NN.
- Implementierung System View (Förderungsstipendium)**
- NN.
- Implementierung HLL View (Förderungsstipendium)**
- NN.
- Anbindung LabVIEW (Förderungsstipendium)**
- Praktikanten, Diplomanden
- (4) Clock-Synchronisation
  - *Univ. Ass. Dr. U. Schmid*  
Leitung  
Konzeption NTSU Hard- und Software
  - *A. Pusterhofer*  
Realisierung NTSU Software (Förderungsstipendium)
  - *Univ. Ass. DI D. Loy* (im Rahmen ESPRIT Eurochip)  
Leitung CSU ASIC Design
  - **V.Ass. NN.**  
**Design und Aufbau CSU ASIC und NTSU Hardware (Dienstvertrag)**

Alles in allem stellte sich die im Projektantrag vorgeschlagene Projektlaufzeit von 2 Jahren (und mit nur zwei Dienstverträgen) sowohl arbeitstechnisch als auch von der Größe der Aufgaben her als ungenügend heraus. Die im Zusammenhang mit der Entwicklung des Prototyps anfallenden Probleme erwiesen sich in der bisherigen Projektlaufzeit doch als umfangreicher als erwartet. Die eben vorgestellte Organisationsstruktur trägt sowohl diesem Punkt als auch den vorgestellten Erweiterungen der Projektziele Rechnung.

Abschließend liegt mir daran, in Anbetracht der doch recht umfangreichen Zusatz-Förderungsmittel noch einmal auf den gleichzeitig vorhandenen hohen wissenschaftlichen Wert und die unmittelbare Praxisrelevanz des Forschungsprojektes hinzuweisen, das somit wohl zu Recht in die als *oriented basic research* bezeichnete Kategorie fällt. Für derartige Projekte ist es ohne Zweifel notwendig, sich nicht mit der Lösung der theoretischen Probleme (die uns innerhalb der projektierten 2 Jahre sicherlich gelingen würde) zufrieden zu geben, sondern den angestrebten Research-Prototypen tatsächlich vollständig zu realisieren; ohne ein "herzeigbares" System dürfte die angestrebte "kommerzielle" Verwertung schwierig sein. Daß die Kosten für diese Realisierung dem FWF schlußendlich auch noch im Rahmen der von der Oesterreichischen Nationalbank geförderten wirtschaftsnahen Projekte refundiert werden, betrachte ich angesichts der von allem Anfang an angestrebten Verbindung von Wissenschaftlichkeit und Praxisorientierung als optimale Konstellation.

### 3. Zusammenstellung der Kosten

Abschließend folgt noch eine kurze Zusammenstellung der beantragten zusätzlichen Förderungsmittel.

|                                                                |                   |
|----------------------------------------------------------------|-------------------|
| <b>Personal:</b> Pos. Nr. 11,12,20-23 und 25:                  | 800.000,--        |
| <b>Wissenschaftliche Geräte:</b> Pos. Nr. 1 - 10,18,19 und 24: | 310.848,--        |
| <b>Sonstige Kosten:</b> Pos. Nr. 15, 16 und 17:                | <u>380.000,--</u> |
| <b>Summe:</b>                                                  | 1.490.848,--      |

## 4. Literatur

- [Bem90] T. Bemmerl, R. Lindhof, T. Treml, *The Distributed Monitor System of TOPSYS*, Proc. CONPAR 90 – VAPP IV Conference, Zürich, 1990, aus: TOPSYS-Tools for Parallel Systems (Artikelsammlung), TU München, Inst. für Informatik, TUM-I9119, SFB-Bericht Nr. 342/13/91 A, Juni 1991.
- [Cou91] D. Couvet, G. Florin, S. Natkin, *A Statistical Clock Synchronization Algorithm for Anisotropic Networks*, Proc. 10th Symp. on Reliab. Distrib. Syst., Pisa, 1991, Real-Time Systems, p. 42-51.
- [Cri89] F. Cristian, *Probabilistic Clock Synchronization*, Distributed Computing 3, 1989, p. 146-158.
- [Hof92] R. Hofmann, R. Klar, B. Mohr, A. Quick, M. Siegle, *Distributed Performance Monitoring: Methods, Tools, and Applications*, Univ. Erlangen-Nürnberg, (submitted manuscript).
- [Kop87] H. Kopetz, W. Ochsenreiter, *Clock Synchronization in Distributed Real-Time Systems*, IEEE Trans. Comput. C-36(8), 1987, p. 933-940.
- [Kop89] H. Kopetz, A. Damm, C. Koza, M. Mulazzani, W. Schwabl, C. Senft, R. Zainlinger, *Distributed Fault-Tolerant Real-Time Systems: The Mars Approach*, IEEE MICRO, Feb. 1989, p. 25-40.
- [Mav92] M. Mavronicolas, *An Upper and a Lower Bound for Tick Synchronization*, Proc. IEEE Real-Time Systems Symposium, Phoenix/Arizona(USA) 1992, p. 246-255.
- [Ol91] A. Olson, K. G. Shin, *Probabilistic Clock Synchronization in Large Distributed Systems*, Proc. 11th int. Conf. on Distributed Computing Systems, Arlington/Texas(USA) 1991, p. 290-297.
- [Pf91] M. J. Pfluegl, D. M. Blough, *Evaluation of a New Algorithm for Fault-Tolerant Clock Synchronization*, Proc. Pacific RIM Symp. on Fault Tolerant Systems, Kawasaki(Japan) 1991, p. 38-43.
- [Ram90] P. Ramanathan, D. D. Kandlur, K. G. Shin, *Hardware-Assisted Software Clock Synchronization for Homogeneous Distributed Systems*, IEEE Trans. Comput. C-39(4), 1990, p. 514-524.
- [Ran91] S. Rangarajan, S. K. Tripathi, *Efficient Synchronization of Clocks in a Distributed System*, Proc. IEEE Real-Time Systems Symposium, 1991, p. 22-31.
- [Ver91] W. A. Vervoort, R. te West, A. L. Schoute, J. Hofstede, *Distributed Time Management in Transputer Networks*, Proc. Euromicro'91 Workshop on Real-Time Systems, Paris-Orsay 1991, p. 224-230.

## A. Liste der bisherige Arbeiten zum Projekt VTA

- [H1] T. Hontsch, *"Source-Code Level Monitoring verteilter Echtzeitsysteme"*, Diplomarbeit TU-Vienna, Department of Automation, Vienna, 1991.
- [H2] T. Hontsch, *"LaSP und ProSP: Funktionsbeschreibung"*, VTA Notes, January 1992.
- [H3] T. Hontsch, *"MUFOM - IEEE Standard for Microprocessor Universal Format for Object Modules"*, VTA Notes, January 1992.
- [HS1] T. Hontsch, U. Schmid, *"Instrumentierung auf Sourcecode-Level"*, Technical Report TU-Vienna, Department of Automation, No. 183/1-26, Vienna, 1992.
- [S1] U. Schmid, *"Monitoring in verteilten Echtzeitsystemen"*, Proceedings Echtzeit '91, Sindelfingen/Germany, 1991, p. 169-176.
- [S2] U. Schmid, *"Monitoring Distributed Real-Time Systems"*, submitted to Real-Time Systems, June 1992.
- [S3] U. Schmid, *"Monitoring of Distributed Real-Time Systems"*, to appear in Proc. NATO Advanced Studies Institute on Real-Time Computing, Sint Maarten, October 1992.
- [SK1] U. Schmid, W. Kastner, *"Monitoring of Distributed Real-Time Systems: The Versatile Timing Analyzer (VTA)"*, to appear in Proc. Workshop Performance Measurement and Visualization of Parallel Systems, Moravany(CSFR), October 1992.
- [SK2] U. Schmid, W. Kastner, *"Konzept eines Distributed Object Based Operating Systems DOBOS"*, VTA Notes, April 1993.
- [SSt1] U. Schmid, S. Stöckler, *"A Versatile Monitoring System for Distributed Real-Time Systems"*, Proceedings Safecomp '92, Zürich, 1992, p. 203-208.
- [St1] S. Stöckler, *"GOLD MINE: Go-ahead Object-oriented Language for Defining Monitors, Events, and Intervals"*, Technical Report TU-Vienna, Department of Automation, No. 183/1-27, Vienna, April 1992.
- [St2] S. Stöckler, *"Timed Attributed Event Traces"*, Technical Report TU-Vienna, Department of Automation, No. 183/1-35, Vienna, April 1993.
- [StSchi1] S. Stöckler, G.H. Schildt, *"Analyse des zeitlichen Verhaltens verteilter Echtzeitsysteme"*, Proceedings Fachtagung Entwurf komplexer Automatisierungssysteme, Braunschweig/Germany, 1992, p. 167-180.