

## Prozeßautomatisierung LU-Test

Automatisierungssystem: Computersystem zur Kontrolle technischer Prozesse. HW muß zuverlässiger, störungsunempfindlicher sein, außerdem umfaßt sie spezielle Prozeßperipherie. SW von SPS bis C-Programmierung unter speziellen Betriebssystemen.

Entwicklungssystem = Targetsystem: historisch gewachsen; Entwicklung auf Targetsystem selbst, alle Entwicklungstools am Target verfügbar - stark aufgeblasenes Targetsystem. NT: unfertige Applikationen können Daten zerstören. VT: Verwendbarkeit von Highlevel-Debuggern.

Entwicklungssystem != Targetsystem: *Crossentwicklung*; VT: physikalische Trennung - unbeabsichtigte Datenzerstörung verhindert. Betriebssystem kann auf spezielle Erfordernisse (geringer Overhead, kurze Reaktionszeiten) zugeschnitten werden. Targetsystem kann während Tests genau wie im späteren Betrieb konfiguriert werden. Highlevelcrossdebugger notwendig - Entwickler merkt gar nicht, daß zu testendes Programm auf Target läuft.

Verbindung Targetsystem - Entwicklungssystem: ICE oder Netzwerke

Target bei LU: Motorola M68030 Mikroprozessor, M-Module-Programmierung in C unter pSOS+

### **Automatisierungssysteme - Hardware:**

störungsunempfindlicher und zuverlässiger und spezielle Prozeßperipherie (digitale/analoge Ein-/Ausgänge = Sensoren/Aktoren). Diese notwendig, weil Größen in technischem Prozeß nicht elektrischer Natur sind (Weg, Temperatur), ebenso Stellbefehle an Aktoren.

#### **1. Aktoren/Sensoren:**

a) **Digitale Sensoren:** verarbeiten diskrete Informationen

- Magnetischer Näherungsschalter: Annäherung von magnetischem Material hat Einfluß auf Magnetfeld. Diese Feldänderung größer als Schwellwert  $\Rightarrow$  Ausgang aktiv
- Induktiver Näherungsschalter: hochfrequenter Oszillator erzeugt mittels Induktivität im frequenzbestimmenden Schwingkreis ein Magnetfeld, das durch die Stirnfläche des Sensors 1-30 mm nach außen dringt. Tritt elektrisch leitender Gegenstand in Magnetfeld  $\Rightarrow$  Wirbelstromverluste  $\Rightarrow$  Dämpfung der Schwingungsamplitude des Oszillators  $\Rightarrow$  Steuerung des Ausgangs
- kapazitiver Näherungsschalter: hochfrequenter Oszillator erzeugt durch seine Kapazität ein elektrisches Feld, das durch die Stirnfläche des Sensors 1-20 mm nach außen dringt. Tritt ein elektrisch nicht neutraler Gegenstand in das Feld  $\Rightarrow$  Frequenzänderung des Oszillators  $\Rightarrow$  Steuerung des Ausgangs
- inkrementeller Drehwinkelgeber: gibt an 2 Ausgängen bei jeder Umdrehung n Rechteckimpulse ab. Auf Achse Codierscheibe (2n lichtdurchlässige/-undurchlässige Segmente), davor fixes Abtastgitter mit 2 Reihen konzentrisch angeordneter, um  $\frac{1}{2}$  Segment versetzte Schlitzen, montiert. Davor LED, deren Licht durch Referenzschlitze und Codierscheibe auf 2 konzentrisch angeordnete Photoelemente fällt. Da Schlitze und lichtundurchlässige Segmente der Codierscheibe deckungsgleich sind werden bei jeder Umdrehung der Achse alle Referenzschlitze einer Spur von Codierscheibe abgedeckt/freigegeben  $\Rightarrow$  Photoelemente liefern Spannungen, diese digitalisiert und als Kanal A,B nach außen geführt. Zusätzliche Referenzmarke und Photozelle für Indesignal Z gibt an bestimmter Winkelposition Impuls ab. Signale A,B um  $90^\circ$  phasenverschoben.  
Beschränkte Genauigkeit  $a,b,c,d = P/4 \pm P/8$ ,  $P/2 \leq l \leq 3P/2$ ,  $P=1/\text{Impulse pro Drehung}$ ,  
Z steigt/fällt wenn B high;  
*Drehung im Uhrzeigersinn*: A,B=LOW  $\Rightarrow$  A=HIGH, B=LOW  
*Drehung im Gegenuhrzeigersinn*: A,B=LOW  $\Rightarrow$  A=LOW, B=HIGH
- Einweglichtschranke: Sender und Empfänger jeweils am anderen Ende 0.1 - 10 m entfernt. Je nach Stärke des Einfalls des modulierten (wegen Fremdlichteinfluß) Lichtes schaltet Ausgang aktiv/inaktiv, wenn größer Schwellwert.

**Hellschaltung:** aktiv bei Lichteinfall

**Dunkelschaltung:** aktiv bei Unterbrechung

- Reflexionslichtschranke: Sender/Empfänger in einem Gehäuse. Modulierter Lichtstrahl zu Objekt (1 - 100 cm entfernt)  $\Rightarrow$  dort diffus reflektiert  $\Rightarrow$  Empfängerlinse sammelt reflektiertes Licht und leitet es an Photoelement weiter. Einfall größer Schwellwert  $\Rightarrow$  Ausgang aktiv. Hell- und Dunkelschaltung möglich
- Glasfaserlichtleiter: zur Erfassung kleiner, schwerzugänglicher Objekte. 2-adrig, am Sender bzw. Empfänger einer Lichtschranke angeschlossen, am anderen Ende offen. Ist Gegenstand nahe genug am offenen Ende  $\Rightarrow$  reflektiertes Licht bewirkt Überschreitung des Grenzwertes  $\Rightarrow$  Ausgang aktiv.

**b) Analoge Sensoren:** zur Erfassung kontinuierlicher Größen

- Distanzsensor auf Triangulationsbasis: LED sendet modulierten Lichtstrahl durch Projektionslinse auf Objekt (1 - 100 cm entfernt). Reflektiertes Licht wird von Empfängerlinse aufgenommen und von Positionselement (PSD), senkrecht zur optischen Achse montiert, registriert. Entfernungsänderung Objekt  $\leftrightarrow$  Sensor ist proportional zu Lichtpunktwanderung auf PSD ( $\Delta b = AB$ ). Diese Änderung wird erfaßt und nach Aufbereitung an Ausgang geliefert.
- Differentialtransformator: zur Messung kleiner Wege (3 - 30 mm). 2 Spulen nebeneinander auf einer Achse angeordnet, drinnen beweglicher Kern. Durch Oszillator hochfrequente Wechselspannung erzeugt, diese in Sendespule eingespeist. In Empfängerspule wird Spannung induziert, deren Amplitude sich mit Eindringtiefe des Kerns ändert. Spannung aufbereitet und an Ausgang geliefert.
- Temperatursensoren:  
höhere Temperaturen - Widerstandfühler: Änderung des Elektrischen Widerstands ändert sich fast linear mit Temperatur  
geringe Temperaturen - Thermoelement: Seebeckeffekt genutzt. Thermospannung zwischen offenen und geschlossenen Enden 2er an einem Ende verschweißter Drähte. Temperaturspannung abhängig von Temperaturdifferenz der offenen/geschlossenen Enden (einige mV)  $\Rightarrow$  Meßumformer, um Signal aufzubereiten.

Linearitätsfehler: erfaßte physikalische Größe und Ausgangsstrom/-spannung nie vollständig linear  $\Rightarrow$  Linearitätsfehler, bei Weiterverarbeitung berücksichtigen.

Stromausgang ist störungsempfindlicher als Spannungsausgang, weil Spannungsabfall in Zuleitung kein Rolle spielt.

Erhöhung der Störungsempfindlichkeit  $\Rightarrow$  differential Outputs (Hin- & Rückleitung) statt single ended Ausgänge (1 Leitung & Ground)

**c) Digitale Aktoren:** für digitale Stellgrößen

- Hubmagnet: mit Gleichstrom betriebener Elektromagnet. Spannung angelegt  $\Rightarrow$  Anker wird in Spule gezogen
- Steppermotor: diskrete Steps möglich (10 - 1000 Steps/Umdrehung). Hohe Positioniergenauigkeit, Drehzahl hoch, aber wegen Massenträgheit trapezförmiges Beschleunigungs-/Abbremsprofil, da sonst Steps verschluckt werden. Aufwendige Ansteuerung, 2 Signale (direction, step).

**d) Analoge Aktoren:** für kontinuierliche Stellgrößen

- Gleichstrommotor: Drehzahl direkt proportional zu angelegter Spannung

## 2. Prozeßinterface:

Aufgabe, von Sensoren gelieferte/an Aktoren gehende Signale an Rechner anzukoppeln. Dadurch kann Rechner Werte lesen/schreiben. Standardinterfaces v.a. Interfacebussysteme (Zwischensystem zw. Aktor/Sensor und Rechnerbus). *Intelligente Aktoren/Sensoren*, die direkt an ein einfaches Netzwerk

angeschlossen werden können, in Rechner nur mehr Netzwerkcontroller, über den Aktoren/Sensoren angesprochen werden können.

### 3. Rechner:

*embedded systems*. Statt zentraler Prozeßrechner immer mehr dezentrale Mikroprozessoren möglichst nahe am Ort der primären Zuständigkeit. Kontrollieren bestimmte Gruppe von Sensoren/Aktoren oder Objekte für Bedienung durch Benutzer.

Verwendet werden: Mikroprozessoren („Rechner“ & Prozeßperipherie auf einem Chip)

SPS bei einfachen Steuerungsaufgaben

Zusammenspiel aller Mikroprozessoren durch geeignete Netzwerkverbindung (Feldbus, Profibus), meist Hierarchie von Rechnern. SPS bei unterstem (prozeßnahem) Level, leistungsfähige Maschinen auf höheren Ebenen.

### Automatisierungssysteme - Software:

Unterschied zu „normaler“ SW: SW für Automatisierungssysteme ist ...

- **prozeßereignisorientiert**: Notwendigkeit zur Aktivität aufgrund externer Ereignisse, meist gleichzeitig, unvorhersagbar, scheinbar unkorreliert
- **funktionskritisch**: bei Fehlfunktion mehr/weniger dramatische Konsequenzen (materieller Schaden, Supergau)
- **zeitkritisch**: Korrektheit von Verarbeitungsergebnissen unabdingbar, ebenso die Rechtzeitigkeit. Einhalten mehr oder weniger harter Zeitschranken.

V.a. Prozeßereignisorientierung fordern *Parallelität* in Echtzeitsystemen. Trend geht zu über Kommunikationssysteme gekoppelte Mikroprozessoren statt großer, teurer Prozeßrechner. Mehrere Prozessoren  $\Rightarrow$  mehr Betriebssicherheit (Fehlertoleranz bei Ausfall von Systemkomponenten), wenn SW dies unterstützt. Aber parallele Prozessoren lösen nicht das Problem der Rechtzeitigkeit (Abhängigkeit von Verarbeitungsschritten untereinander). Time constraints!

- **statische EZS**: basieren auf Annahme der Vorhersagbarkeit von Ereignissen. Aktivitäten gehen vom Rechner aus, time driven (Verarbeitung zu bestimmter Uhrzeit). Polling (zyklisches Abfragen). VT: Einhaltung best. Zeitbedingungen a priori garantiert, Fehlertoleranzmechanismen relativ leicht zu implementieren. NT: Zeit zwischen den Messungen muß klein genug sein (schwierig bei schnellen, instabilen Prozessen, ebenso bei dynamisch variierenden Betriebsbedingungen)  $\Rightarrow$  worst case Abschätzung  $\Rightarrow$  Überdimensionierung der HW. Ressourcen teuer, groß, schlecht ausgenutzt im Normalfall. Aufwand meist nur bei hochzuverlässigen Anwendungen vertretbar.
- **dynamische EZS**: Aktivität geht vom technischen Prozeß aus, event driven. NT: keine a priori Garantie für Einhaltung von Zeitschranken, Fehlertoleranzmechanismen schwer zu implementieren. VT: bessere Auslastung der HW-Ressourcen, organische Anpassung an nichtdeterministische techn. Prozesse.

### Entwicklung von Automatisierungssystemen:

techn. Prozeß  $\rightarrow$  *Anforderungsspezifikation* (Analyse des technischen Prozesses)  $\rightarrow$  *Lösungsansätze*  $\rightarrow$  HW & SW-Lösung: HW meist fix durch Standardkomponenten, Baukastensystem der Module, nicht unmittelbar zu realisierende Anforderungen  $\rightarrow$  Spezialentwicklung. HW-Lösung & Anforderungsspezifikation ist Grundlage für SW-Lösung: Es gibt Standard-SW aller Komplexität (vom nackten Kernel bis zu komplexen Systemen plus Tools). Meist unterstützt SW eine HW-Familie völlig  $\rightarrow$  problemspezifische Lösung nur bei Applikations-SW  $\rightarrow$  Abbildung des Problems auf SW/HW-Standard  $\rightarrow$  Aufteilung in Tasks, Festlegung der Synchronisations- bzw. Kommunikationserfordernisse. Realisierung SW-Schnittstelle  $\leftrightarrow$  Prozeßperipherie (Interrupt-Handling).  
*Implementierungsphase*: zeitkritische Interrupt-Handler in Assembler, wegen beschränkter HW (Speicherplatzmangel) wird auf Utilityfunktionen meist verzichtet  $\Rightarrow$  niedriges Implementierungsniveau.  
*Testphase*: Qualitätsanforderungen an HW/SW wegen Folgen höher, Test am lebenden Objekt erst spät vertretbar  $\rightarrow$  Simulation des technischen Prozesses parallel zur Entwicklung des eigentlichen Produktes.

*Wartung:* da Produktionsausfälle teuer umfangreiche Diagnosetools zum Aufspüren von HW-/SW-Fehlern. Behebung von SW-Fehlern soll rasch durchführbar sein, ebenso Installation neuer Versionen, Fernwartung!

### Bedienpanel Simple I/O:

- *BCD:* 4 Leitungen (1,2,4,8) gegen Ground durchgeschaltet
- *7-Segmentanzeige:* 7 rechteckige Leuchtdioden, Dezimalpunkt und Summer gemeinsam, Anoden (+) der LED's einzeln, Kathoden (-) zusammengeschaltet
- *Summer:* beim anlegen von Gleichspannung Tonabgabe, zusammen mit Dezimalpunkt 7-Segmentanzeige angesteuert

### Aufbau Floppyteststand: von links nach rechts

- Gleichstrommotor
- Schlitten, darauf Floppylaufwerk montiert
- Floppyschließer
- automatischer Floppystopfer
- programmierbarer Floppyschließer
- Barcodeleser
- Teststation

### Sicherheitseinrichtungen / Handsteuerung: nicht per SW beeinflussbar

- *mechanische Endeschalter:* verhindern Anschlagen des Schlittens linke/rechte Endeposition. Spricht einer der beiden Endeschalter an  $\Rightarrow$  Schlittenmotor hw-mäßig ausgeschaltet
- *Handsteuerungstaster:* Schlitten mit 80% Maximalgeschwindigkeit links/rechts fahrbar. Handsteuerung überlagert Rechnersteuerung
- *automatischer Floppystopfer:* soll Schließen des Laufwerks verhindern, wenn Floppy noch nicht vollständig im Laufwerk eingerastet ist.

### Memory Mapped I/O:

Zugriffe auf Speicher und Peripherie sind gleich. Register einer Karte werden können als „Speicherbereich“ gesehen werden, der an einer fixen Adresse liegt. Die Registerstruktur wird mittels „struct“ definiert und über Pointer referenziert. Dieser Pointer muß auf die Basisadresse des Speicherbereiches zeigen.

a) **Digitale Outputs:** binäres Signal: log 0  $\rightarrow$  0V (low); log 1  $\rightarrow$  5V (high)

Output-Port (*write only*): Port  $\leftrightarrow$  Verbindung zum technischen Prozeß im Vordergrund steht. Aus der Sicht des Prozessors Output Register. Register hat hw-mäßig fixe Adresse zugeordnet (!= gültiger Speicheradresse)

Schreiboperation: 4-Bit-Datenwort auf Datenbus  $\Rightarrow$  Zieladresse auf Adreßbus und notwendigen Signale an Kontrollbus (read - log 0). Adreßdekoder vergleicht an Adreßbus anliegende Adresse mit seiner, gibt gegebenenfalls log 1 auf Leitung adress aus. Schreibender Zugriff legt log 1 an Clockeingänge D-Flipflop, diese übernehmen an Datenbus anliegendes Bitmuster an Ausgänge  $\rightarrow$  dieses bleibt bis zum nächsten Schreibzugriff des Prozessors im Register stehen.

b) **Digitale Inputs:**

Input-Port (*read only*): ebenso eindeutige Speicheradresse zugeordnet (!= gültiger Speicheradresse)

Leseoperation: Adresse auf Adreßbus und notwendigen Signale an Kontrollbus (read - log 1). Adreßdekoder vergleicht  $\rightarrow$  log 1, AND-Gatter liefert log 1 an 4 parallelgeschaltete Output-Enable-Eingänge der Tristate-Buffer, die an Inputleitungen anliegendes Bitmuster auf Datenbus durchschalten.



**Data Direction Register:** Port so realisiert, daß Leitungen sowohl als Input als auch als Output verwendet werden können  $\Rightarrow$  Data Direction Register für Konfiguration der HW notwendig. Jedes Bit des Registers ist Leitung des Ports zugeordnet. Ist es 1  $\Rightarrow$  Leitung Input, sonst Output

### c) Analoge Outputs:

Analoge Signale sind nicht nur diskrete Werte, sondern jeder beliebige Zwischenwert kann vorkommen. Realisiert mittels *D/A-Wandler* = elektr. Schaltung, die an einem analogen Ausgang jene Spannung erzeugt, die der in Register geschriebenen Binärzahl entspricht.

Kenngößen: Auflösung, Konversionszeit, 12 Bit D/A-Wandler,  $\pm 10$  V  $\Rightarrow 20/4096 = 4.88$  mV

### d) Analoge Inputs:

*A/D-Wandler* ermitteln zu analogem Spannungswert die Binärzahl. Diese an analogen Input gestellt, dort von Prozessor lesbar aus Register;

Auflösung (w.o. 4.88 mV, falls Werte sich um weniger als 4.88 mV unterscheiden  $\Rightarrow$  derselbe binäre Wert). A/D-Wandlung ist zeitaufwendig, daher wird sie explizit gestartet und nicht kontinuierlich. A/D-Wandler mit hoher Auflösung sind teuer und komplex  $\Rightarrow$  mehrere Inputs  $\Rightarrow$  Analogmultiplexer: Diese schalten selektiv den Wandler auf einen der analogen Inputs. Vor dem Start muß Multiplexer Adresse des Kanals gegeben werden. Alle Eingänge können bedient werden, aber nicht gleichzeitig.

**Signal conditioning:** Aufgabe, den tatsächlichen Eingangsbereich ( $\pm 10$  V, 4-20 mA) auf den für den A/D-Wandler notwendigen Bereich abzubilden. Außerdem Filterung, um Störsignale zu unterdrücken.

## VME-Komponenten:

- **Carrier Board I A201:** Carrierboard für 4 M-Module (M3, M21\_B, M28, M21\_T). Koppelt einfachen Interfacebusse der M-Module an komplexen VME-Bus. Die auf den M-Modulen befindlichen Register werden in den VME-Adreßraum gemapped, so daß sie mittels normaler VME-Schreib-/Lesezyklen angesprochen werden können. Log. Adreßraum der M-Module in 0x100 Byte große Fenster in VME-Adreßraum. Control-/Vektorregister nicht auf M-Modulen, sondern auf A201 für Interrupt-Handling. Jedem Interrupt wird eindeutiger Interruptvektor zugeordnet, dieser in Vektorregister eingetragen. Manche Module haben kein Vektorregister (M21), können aber Interrupts auslösen. Andere (M331) haben mehrere Interruptquellen/-register  $\Rightarrow$  A201 programmierbar ausgelegt. Für jedes Modul 0-3 gibt es eigenes Control-/Vektorregister, damit kann der von den Modulen unterstützte Interruptmode programmiert werden (Interrupt extern: M331; Interrupt intern: M21)
- **Carrierboard II A201:** Carrierboard für 4 M-Module (M4, M3, M331, M333).
- **Transistor Output Module M3:** 16 optogetrennte, strombegrenzte open-collector Ausgänge. Last gegen Ground (*sink output*)  $\Rightarrow$  M3 invertiert (Bit == 1  $\Rightarrow$  Ausgang gegen Ground). 16 Bit Outputregister, Zugriff: Bit, Byte, Word. *BIG ENDIAN* bei Byteaccess!
- **Transistor Output Module M28:** 16 optogetrennte, strombegrenzte open-emitter Ausgänge. Last gegen Versorgungsspannung (*source output*, Bit == 1  $\Rightarrow$  Ausgang gegen Versorgungsspannung). 16 Bit Outputregister, Zugriff: Bit, Byte, Word.
- **Binary Input Module M21:** 16 optogetrennte digitale Eingänge. Diese sind Konstantstromquellen (*sourcend*), die direkten Anschluß eines gegen Ground schaltenden Schalters erlauben (*sink load*). Eingänge sind hw-mäßig *entprellt* (Änderung erst gültig, wenn 5.5 ms neuer Zustand stabil bleibt). In diesem Fall wird Interruptleitung der M21 aktiviert. Ob Interrupt ausgelöst wird, ist abhängig von der Programmierung des Controllregisters der A201. Ausgelöster Interrupt wird während Interrupt Acknowledge Cycles zurückgesetzt. Bit == 0  $\Rightarrow$  Eingang auf Ground. 16 Bit Inputregister, Zugriff: Bit, Byte, Word.
- **12 Bit Analog Input M-Module M34-1:** 16 singleended, optogetrennte analoge Eingangskanäle (Auflösung 12 Bit, Konversionszeit 10  $\mu$ s). Eingangsverstärkung und Betriebsart(unipolar/bipolar) programmierbar, Eingangsbereich  $\pm 10$  V. M34-1 unterstützt asynchrone Konversion, Autoinkrementierung der Eingangskanalnummer des Analogmultiplexers. Dataregister hat

konvertierten Spannungswert (bipolarer Fall) in **Zweierkomplementdarstellung**. In Moderegister geschriebene Konfiguration erst NACH einer Konversion aktiviert.

- **12 Bit Analog Output M4:** 4 optogetrennte, analoge Ausgangskanäle (Auflösung 12 Bit, Konversionszeit 10 µs). Ausgangsverstärkung und Betriebsart (unipolar/bipolar) programmierbar. In Bipolarmode auszugebende Spannung in **Exzeßdarstellung**

## Struktur eines SW-Systems unter pSOS<sup>+</sup>:

- **System-SW** = alle zum pSOS<sup>+</sup>-Betriebssystem gehörenden Teile
  - Kernel (pSOS<sup>+</sup>)
  - System-State-Debugger (pROBE<sup>+</sup>, für Kommunikation mit XRAY)
  - TCP/IP-Support (pNA<sup>+</sup>)
  - C-Library (pREPC<sup>+</sup>)
  - Configuration Tables (um pSOS<sup>+</sup>-Komponenten an spezielle HW anzupassen)
 Alle Teile in Eprom auf Target untergebracht!
- **Tasks:**  
 Applikations-SW besteht aus Tasks, die über Queues, Semaphore, Events, globale Variablen kommunizieren. Verwaltung des Tasks ist Sache des Betriebssystems. *IDLE-Task* wird immer dann exekutiert, wenn keine andere Task lauffähig ist, verbraucht nur CPU-Zeit, von pSOS<sup>+</sup> automatisch erzeugt. *ROOT-Task* wird nach HW-Reset von internen Initialisierung gestartet. Dieser startet Applikations-SW. Besteht diese aus mehreren Tasks ⇒ diese in *ROOT-Task* explizit starten. *Main()* ist nur Funktion, die in *ROOT-Task* ohne Parameter aufgerufen wird.
- **Interrupt Service Routinen:**  
 Prozeßperipherie so ausgelegt, daß die meisten Eingänge Interrupts auslösen können. Interrupt bewirkt Programmunterbrechung und Ausführung einer Interrupt Service Routine, die diesem Interrupt zugeordnet ist. ISR sollen nur die notwendigen Aktionen durchführen, da *zeitkritisch*. Für eigentliche Reaktion auf Ereignis bestimmte Tasks aktivieren. Bestimmte Mechanismen zur Intertaskkommunikation auch aus ISR verwendbar.
- **Device Driver:**  
 pSOS<sup>+</sup> unterstützt Device Driver Konzept (vgl. Unix). Terminal wird über solchen Driver gesteuert. Meist sind Device Driver für Aufgabenstellung an Automatisierungssystem nicht brauchbar.

## VME-Bus:

VT:

- nicht gemultiplexer, asynchroner Datentransfer
- mehrprozessorfähig
- 4 priorisierte Busanforderungsebenen
- 7 Interruptebenen
- Unterstützung schneller Fehlererkennung

### 2 Subsysteme:

- VMS-Bus (Datenaustausch innerhalb Baugruppen)
- VSB

### 3 Subsysteme:

- Master
- Slave
- Controller

*Master* (z.B. CPU-Karte) fordert VME-Bus aktiv an, bestimmt Richtung des Datenflusses, kann an jedem Steckplatz außer Nr1 stecken.

*Slave* (z.B. I/O-Karte) wird von Master angesprochen. Ausnahme: Interruptanforderung

Bei mehreren Mastern muß *Arbiter* vorhanden sein (pro VME-Bussystem nur 1 Arbitr). Dieser sitzt auf Steckplatz 1 der Controllerbaugruppe. Bei mehr als 4 Master *Daisy chaining* (Priorität über physikalischen Einbauplatz)

#### 4 Bereiche:

- Datentransferbus
- Interruptbus
- Arbitrationbus
- Utilitybus

Wird von Slave innerhalb bestimmter Zeit oder nie quittiert, kann Master Buszyklus nicht beenden ⇒ System hängt sich auf. *Watch Dog Funktion* verhindert dies, in dem dann Busfehlersignal aktiviert wird (*Bus timer*)

#### Optimierende Compiler:

Beim Schreiben einer Outputregisters/Lesen eines Inputregisters muß man auch Auswirkungen von/nach außen berücksichtigen. Doppelte Zuweisung würde ohne Deklaration mittels 'volatile' wegoptimiert werden.