Problemlösung mit Mikrocontrollern

- Ein Kurzüberblick -

Mikrocontroller sind programmierbare Universalrechner, die zum Einsatz in Embedded Systems optimiert sind. Kennzeichnende Merkmale:

- Implementierung der jeweiligen Informationswandlungen auf Grundlage herkömmlicher Programmmiermodelle,
- Kostenoptimierung,
- Einbau in eine Anwendungsumgebung,
- Programmierung auf den jeweiligen Anwendungszweck hin (Controller wird durch Programmierung zur Einzweckmaschine),
- Programme typischerweise vom Nutzer nicht änderbar,
- wir dürfen programmieren (Narrenfreiheit, potentielle Funktionsvielfalt),
- wir müssen programmieren (Zwang zur Rückführung der anwendungsseitig geforderten Informationswandlungen auf elementare Programmschritte; Serialisierung der Informationsverarbeitung, Problemlösung entspricht nicht mehr dem anwendungsseitigen Datenflußschema).

Anwendungsfälle:

- Realzeitraster im ms-Bereich oder langsamer,
- es stehen keine anderweitigen Forderungen entgegen (extremes Stromsparen, EMV, Sicherheitsvorschriften).

Mikrocontroller sind meist deutlich kostengünstiger als CPLDs und einfacher zu programmieren.

Infrastruktur:

- Stromversorgung,
- Takt,
- Rücksetzen,
- E-A-Anschaltung,
- ggf. Speicheranschaltung.

Vorsicht vor Buskonflikten beim Einschalten und im Betrieb!

Hochohmige (floating) Busleitungen
Vermeiden. (Pull-up-Widerstände, Bushalteschaltungen, Park-Hardware.)

Port-Belegung

- Im Problemfall Ports stets für die höher integrierte Funktion im µC ausnutzen, dafür die einfachen I/O-Funktionen auslagern.

ungenutzte Ports:

- als Ausgänge und mit Festwert belegen (nur, wenn garantiert unbeschaltet)
- als Eingänge mit externem Pull-up
- als Eingänge mit internen Pull-up's (diese ggf. scharfmachen)
- NIE als wirklich offene Eingänge (falls nicht ausdrücklich im Datenblatt als zulässig spezifiziert)

Bus-Erweiterung mit Pufferstufen und Registern

- Buskoppelstufen und Register
- Abfrage: Multiplexer
- Universal-Busschaltkreise ausnutzen
- Boundary Scan
- einfache Schiebe-Interfaces

Auf *asynchrone Eingänge* achten (Metastabilität). Ggf. über D-FF-Register synchronisieren.

Taktierung bzw. Output Enable von außen - nur selten vorhanden ("echte" Tristate-Ports, um µC seinerseits als Slave an einen Bus zu hängen)

Bei Anschaltung von Bustreibern *Buskonflikte* vermeiden (Richtungssteuerung).

Speicheranschaltung

dafür vorgesehene Prozessoren

- ausgebauter Speicherbus (H8/300H)
- Multiplex-Speicherbus (PIC, 8051 usw.)

sonst: Ports für Anschaltung ausnutzen. Es gibt auch spezielle kleine RAMs (z. B. von Siemens).

Adreßdecodierung
Alias-Adressen: nie für Programmiertricks ausnutzen!

Adreßerweiterung
Bankregister (oder Segment-RAM) und Bank-Organisation. Heutzutage: einfachste Hardwarelösung verwenden. (Genügt diese nicht, anderen Prozessortyp wählen).

Externe Beschleunigungsmaßnahmen

- Adresse für Ausgabezwecke ausnutzen
- Parallelisierung von Speicher- und Interfacezugriffen
- Speicher breiter gestalten, als Art Mikroprogrammspeicher betrieben (µC übernimmt Sequencing)
- Aufschalten von Daten auf Teile des Datenbus
- externe Steuer- bzw. Zwischenspeicher (adressierbar oder FIFO)

Einfache Multiprozessorkopplungen

- Direktverbindungen
- Dual-Port-RAMs
- einfache Bussysteme
- Schiebering-Strukturen
- geschaltete Verbindungen (Crossbar, Switched Fabric)

Externe Port-Hardware

Grundlagen der Auswahl - Wofür?

- funktionelle Anpassung,
- Anpassung der Zugriffsbreiten,
- Anpassung an das Zeitverhalten (Timing),
- Pegelwandlung,
- Erhöhung der Treibfähigkeit,
- Isolation (Partial Power Down, Hot Plugging),
- Interface zu Leistungs- oder Analogschaltungen.

Besonderer Probleme des Tri-State-Prinzips

- Konfliktfälle (Bus Contention) vermeiden,
- Gefahr von Bus Contention beim Ein- und Ausschalten (erfordert gelegentlich Voltage Monitor zur Rücksetzsignal-Erzeugung),
- der Tri-State-Bus in Ruhe. Schwimmende Pegel vermeiden: Pull-up- oder Pull-down-Widerstände, Bus-Hold-Schaltungen (zusätzlich oder eingebaut), Parken.

Kochbuchmäßige Anschaltung an echte Tri-State-Bussysteme (z. B. dem Speicherbus der typischen Controller o. Prozessoren (8051 usw.)) zumeist problemlos.

Vorsicht aber bei programmierter Steuerung eines Tri-State-Bus. Richtig initialisieren! Umschalten nach Prinzip Break before Make!

Treiber bzw. Puffer (Drivers, Buffers)
Treiber bzw. Puffer sind Schaltstufen ohne Speicherglieder. Die Signale werden nur weitergereicht (und dabei ggf. invertiert). Es gibt aber keine logischen Verknüpfungen von Signalen.

Typische Anwendungen:

- zum Aufschalten von Eingangssignalen (z. B. auf einen Datenbus),
- zum Liefern von Ausgangssignalen mit entsprechender Treibfähigkeit,
- zur Pegelwandlung,
- zum Entkoppeln in Betriebsfällen des Partial Power Down (Stromsparen, Hot Plugging).

Unidirektionale und bidirektionale Treiber/Puffer

- unidirektional = nur eine Signalflußrichtung (vom Eingang zum Ausgang),
- bidirektional = Signalflußrichtung umschaltbar.

Industriestandard-Typen unidirektionaler Treiber/Puffer

125, 126

4 Bitpositionen. Nicht invertierend. Jeweils einzeln steuerbar.

- 125: Steuersignale aktiv Low,
- 126: Steuersignale aktiv High.

Vorzugstyp: 125. Auch als Single Gate (nur 1 Bitposition).

240, 241, 244
8 Bitpositionen in 2 4-Bit-Gruppen. Jede Gruppe einzeln steuerbar.

- 240: invertierend, beide Steuersignale aktiv Low,
- 241: nicht invertierend, das eine Steuersignal aktiv Low, das andere aktiv High
- 244: nicht invertierend, beide Steuersignale aktiv Low.

365...368
6 Bitpositionen, 2 Steuersignale (beide aktiv Low). Nur HC-Baureihen.

- 365: nicht invertierend. Gemeinsame Aufschalterlaubnis über konjunktive Verknüpfung beider Steuersignale.
- 366: invertierend. Aufschalterlaubnis wie 365.
- 367: nicht invertierend. Das eine Steuersignal wirkt auf 4, das andere auf 2 Signalwege.
- 368: invertierend. Aufschalterlaubnis wie 367.

540, 541
8 Bitpositionen. 2 konjunktiv verknüpfte Steuersignale (beide aktiv Low).

- 540: nicht invertierend,
- 541: invertierend.

Vorzugstypen: 240, 244.

Multiplexer mit Tri-State-Ausgängen
Anwendung: zum Aufschalten auszuwählender Eingangssignale. Es kommen nur Typen mit Tri-State-Ausgang in Frage.

- 257: 4-fach 2-zu-1, nicht invertierend,
- 258: 4-fach 2-zu-1, invertierend,
- 857: 6-fach 2-zu-1, nicht invertierend,
- 253: 2-fach 4-zu-1, nicht invertierend,
- 251: 8-zu-1, 2 komplementäre Ausgänge.

Vorzugstypen: 257, 253.

Vorteil der Multiplexer: keine Konflikte beim Abfragen. Gelegentlich Nachteil: keine “busmäßige” Leiterzugführung.

Industriestandard-typen bidirektionaler Koppelstufen (Transceivers)

Anwendung: zum Entkoppeln bzw. Puffern bidrektionaler Datenwege. Bei fest beschalteter Richtungssteuerung auch als unidirektionale Treiber/Puffer verwendbar.

245
8 Bitpositionen, nicht invertierend. 1 Richtungssteuer- und ein Aufschalterlaubnissignal. Aufschalterlaubnis aktiv Low. *Der* Industriestandard (alle anderen Typen kommen praktisch nicht mehr in Frage). B-Seite hat in manchen Baureihen eine höhere Treibfähigkeit.

620
8 Bitpositionen, nicht invertierend. Je ein Aufschalterlaubnissignal für jede Richtung (das eine aktiv Low, das andere aktiv High). Beide Aufschaltsignale dürfen gleichzeitig aktiv sein (Betriebsweise der Rückkopplung bzw. Selbsthaltung (Bus- Hold-Funktion)).

623
Wie 620, aber nicht invertierend.

640
wie 245, aber invertierend.

Register
Register enthalten Latches oder flankengesteuerte Flipflops.

Typische Anwendungen:

- zum Zwischenspeichern und Aufschalten von Eingangssignalen (z. B. auf einen Datenbus). Erfordert Typen mit Tri-State-Ausgängen.
- zum Zwischenspeichern und Liefern von Ausgangssignalen mit entsprechender Treibfähigkeit,
- Adreßhalteregister bei Speicheranschaltung an Multiplex-Bussysteme (z. B. 8051).

Latch oder Flipflop?

- gemäß Kochbuch (entsprechend Bussystem des Controllers/Prozessors) oder gemäß Zeitverhalten auswählen.
- bei programmierter Taktimpulsbildung aufpassen. Latches folgen Änderungen der Datenbelegung nach, solange Takt aktiv ist.

Metastabilität

- im Datenmaterial nachlesen.
- zum direkten E-A-Anschluß vorgesehene Ports von Mikrocontrollern sind typischerweise Metastable-resistant, nicht aber alle Bussysteme (je leistungsfähiger der Prozessor, desto weniger ist damit zu rechnen).
- Zum Eintaktieren asynchroner Signale keine Latches verwenden.

Industriestandard-Typen

Latches:

- 373: 8 transparente Latches mit Tri-State-Ausgängen. Nicht invertierend. Takt aktiv High, Aufschalterlaubnis aktiv Low.
- 573: wie 373, nur andere Anschlußbelegung.
- 533: wie 373, aber invertierend.

D-Flipflops:

- 374: 8 D-Flipflops mit Tri-State-Ausgängen. Nicht invertierend. Datenübernahme mit Low-High-Flanke des Taktes, Aufschalterlaubnis aktiv Low.
- 574: wie 374, nur andere Anschlußbelegung.
- 534: wie 374, aber invertierend.
- 273: 8 D-Flipflops mit binären Ausgängen (kein Tri State!). Nicht invertierend. Datenübernahme mit Low-High-Flanke des Taktes. Zusätzliches Löschsignal (aktiv Low), auf alle Flipflops wirkend.
- 377: 8 D-Flipflops mit binären Ausgängen (kein Tri State!). Nicht invertierend. Gemeinsames Erlaubnissignal (aktiv Low) zur Steuerung der Datenübernahme,

Selbsthaltung, wenn inaktiv (Verhalten ≙ DV-Flipflop (voll synchron)). Datenübernahme mit Low-High-Flanke des Taktes bei aktivem Taktsteuersignal.

Vorzugstypen: 373/573, 374/574.

Bidirektionale Register

- 543: 8 Bitpositionen. Datenweg nicht invertierend. Für jede Richtung ein Latch-Register. 2 unabhängige Richtungssteuersignale. Je Richtung ein Aufschalterlaubnis- und ein Taktsignal. Alle diese Signale aktiv Low wirkend.
- 544: wie 543, aber Datenweg invertierend.

Treiber, Puffer und Register mit Serienwiderständen (Output Damping Resistors)

Schaltkreise enthalten in den Ausgangswegen Serienwiderstände (typisch 25 Ω). Nutzung: Leitungsanpassung (vor allem bei Speicheransteuerung), Strombegrenzung, Flankenverschleifung (Stichwort: EMV).

Register mit Einzelbitzugriff

259
8 Latches, wovon jeweils eines adressierbar ist. 8 Ausgänge, binär (kein Tri State!) , nicht invertierend. 3-Bit-Adresse, 1 Dateneingang, 1 Übernahmeeingang (= Takt, aktiv High), 1 Löscheingang (wirkt aktiv Low auf alle 8 Latches).

Zurücklesen von Registern

Grundsatzlösungen:

- Halten von RAM- oder Register-Kopien mittels Software (Registerzugriffe nur über Unterprogramme o. Macros),
- automatisches Halten von RAM-Kopien: gleichzeitige E-A-und Speicherzugriffe. Schaltungsaufwand.
- zusätzliche E-A-Ports oder Treiberschaltkreise zum Zurücklesen,
- rücklesbare Register (Readback Latches),
- entsprechende Ausnutzung von Mehrzweckschaltkreisen (Universal Bus Transceivers).

Rücklesbare Register (Readback Latches):

- 666: 8 Latches, gemeinsam löschbar, Ausgänge nicht invertiert,
- 667: wie 666, aber invertierte Ausgänge,
- 990: 8 Latches, Ausgänge nicht invertiert,
- 991: wie 990, aber invertierte Ausgänge,
- 992: 9 Latches, Ausgänge nicht invertiert,
- 993: wie 992, aber invertierte Ausgänge,

- 994: 10 Latches, Ausgänge nicht invertiert,
- 995: wie 994, aber invertierte Ausgänge,
- 996: 8 D-Flipflops, wählbare Polarität der Ausgänge.

Nachteile:

1. Ausgangsbelegung am Schaltkreis-Anschluß ist nicht zurücklesbar,
2. Exoten: vergleichsweise teuer, keine modernen Gehäuse, nur 5-V-Typen, fragliche Zukunftssicherheit.

Koppelstufen und Register mit 9 oder 10 Bitpositionen

- 821: 10 D-Flipflops mit Tri-State-Ausgängen. Wie 574.
- 823: 9 D-Flipflops mit Tri-State-Ausgängen. Nicht invertierend. Aufschalterlaubnis aktiv Low. Gemeinsames Taktsteuersignal (aktiv Low; wirkt auf die Takteingänge der Flipflops). Datenübernahme mit Low-High-Flanke des Taktes bei aktivem Taktsteuersignal. Zusätzliches Löschsignal (aktiv Low), auf alle Flipflops wirkend.
- 827: Treiber/Puffer mit 10 Bitpositionen. Wie 540.
- 841: 10 Latches mit Tri-State-Ausgängen. Wie 573.
- 843: 9 Latches mit Tri-State-Ausgängen. Wie 573. Zusätzlich ein Setz- und ein Löschsignal, beide aktiv Low auf alle Latches wirkend.
- 861: bidirektionaler Treiber/Puffer mit 10 Bitpositionen. Wie 620, aber beide Aufschalterlaubnissignale aktiv Low.
- 863: bidirektionaler Treiber/Puffer mit 9 Bitpositionen. Wie 620, aber je Richtung 2 Aufschalterlaubnissignale in konjunktiver Verknüpfung (sämtlich aktiv Low).

Achtung - Taktsteuersignale:

- auf Datenweg wirkend (wie DV-Flipflop; vgl. Typ 377): den gleichen Setup- und Haltezeitspezifikationen und Metastabilitätsbedingungen unterworfen wie die Daten,
- auf Takt wirkend: Steuerung über Latch-Kreis. Besondere Setup- und Haltezeitspezifikationen beachten. Metatsabilitätsverhalten kritisch.

Mehrzweckschaltkreise (Universal Bus Transceivers)

Derartige Schaltkreise lassen sich für verschieden Signalflußrichtunen sowie Betriebsarten mit und ohne Speicherung einsetzen. Kennzeichnend sind eine Vielzahl von Steuersignalen sowie eine recht komplizierte Innenschaltung.

- 646: 8 Bitpositionen, 2 D-Flipflops je Bitposition. Datenweg nicht invertiert. Gemeinsames Richtungssteuersignal.
- 648: wie 646, aber Datenweg invertiert.

- 651: 8 Bitpositionen, 2 D-Flipflops je Bitposition. Datenweg invertiert. Ein Richtungssteuersignal je Datenweg.
- 652: wie 651, aber Datenweg nicht invertiert.
- 2952: wie 652. Zusätzliches Taktsteuersignal für jede Richtung.
- 16500: 18 Bitpositionen, 2 Speicherglieder je Bitposition, die sich wahlweise als D-Flipflop, als Latch oder als Durchreiche schalten lassen (gemeinsame Betriebsartenwahl für alle Bitpositionen). Taktsignale aktiv Low (Latch) bzw. mit High-Low-Flanke (D-Flipflop) wirkend.
- 16501: wie 16500, aber Taktsignale aktiv High (Latch) bzw. mit Low-High-Flanke (D-Flipflop) wirkend.
- 16600: wie 16500, aber zusätzliche Taktsteuersignale (aktiv Low).
- 16601: wie 16501, aber zusätzliche Taktsteuersignale (aktiv Low).
- 16524: 18 Bitpositionen. Signalweg in die eine Richtung wahlweise über eine 4-stufige Pipeline aus D-Flipflops oder über ein D-Flipflop-Register, Signalweg in die andere Richtung ist direkte Durchreiche.
- 16525: wie 16524, aber Signalweg in 2. Richtung über D-Flipflop-Register.
- 32316: 3 bidirektionale Datenwege zu je 16 Bitpositionen. Je Datenweg ein D-Flipflop. Schaltbare Signalflüsse je Datenweg: (1) Eingabe in das “eigene” D-Flipflop, (2) Ausgabe aus einem der D-Flipflops der anderen Datenwege (wählbar).
- 16260: 12-Bit-auf-24-Bit Multiplexer/Demultiplexer mit Latches.
- 16268: 12-Bit-auf-24-Bit Multiplexer/Demultiplexer (Bus Exchanger) mit D-Flipflops. Vorzugsweise zum Demultiplexen (vom schmalen zum breiten Bus).
- 1629: 12-Bit-auf-24-Bit Multiplexer/Demultiplexer (Bus Exchanger) mit D-Flipflops. Vorzugsweise zum Multiplexen (vom breiten zum schmalen Bus).

Breite Ausführungen (Widebus)

Typische Breiten: 16 - 18- 20 - 32- 36 Bits. Kleine Gehäuse, geringe Anschlußabstände. Praktisch nicht auf 2-Ebenen-Leiterplatten einsetzbar. Empfindlich gegen Bus Contention beim Ein- und Ausschalten.

Adreßtreiber und -Register

Schaltkreise verteilen jeweils 1 Bit auf 2 oder 4 Ausgänge, die einzeln steuerbar sind. Gelegentlich als Verteiler brauchbar. Geringer Skew!

- 16830: 18 Bitpositionen. Je Bitposition werden 2 Ausgänge getrieben. 2 Aufschalterlaubnissignale, jeweils auf eine Reihe von Ausgängen wirkend.
- 16831: 9 Bitpositionen. Je Bitposition werden 4 Ausgänge getrieben. 2 Aufschalterlaubnissignale, jeweils auf zwei Reihen von Ausgängen wirkend. Durchschaltung wahlweise kombinatorisch oder über D-Flipflop (eines je Bitposition).
- 16832: wie 16831, aber nur 7 Bitpositionen.

Pegelwandlung

Entkopplung (Partial Power Down)

Logikbaureihen

Welche Baustufen gibt es in welcher Technologie für welche Speisespannung?

Schieberegister-Interfaces

Boundary Scan

E-A-Schaltungen in programmierbarer Logik

Programmierbare Logik oder Industriestandard?
Programmierbare Logik wesentlich teurer und zumeist weniger robust. Erlaubt aber Einbau weiterer Funktionen, z. B. Kombinatorik zwecks Entlatsung des Mikrocontrollers. Bei CPLDs auf Verhältnis von Zellen und Pins achten. Typischerweise keine gute Auslastung, wenn 1 Zelle ≈ 1 Pin (Xilinx 9500).

Welche E-A-Funktionen müssen im fertigen System noch programmierbar sein?

- Im fertigen Systementwurf liegen die Verbindungen fest (was ist Input, was Output, was bidirektional). Die meisten Datenwege sind letzen Endes aus funktioneller Sicht starr (hard wired).
- Vollkommen freie Programmierbarkeit der Ports im Mikrocontroller nur wegen der Universalität der Schaltkreise. Nur selten zwecks Flexibilität in der Anwendungsumgebung genutzt. Ports werden typischerweise nur beim Initialisieren eingestellt.

Direkt koppelbare Leistungsschaltungen

Ähnlich Industriestandard
Open-Drain-Ausgänge für Low Side Drive.

- TPIC6259/6A259 ≙ 259 - 8 einzeln stellbare Latches,
- TPIC6273 ≙ 273 - 8 D-Flipflops, gemeinsam löschbar,
- TPIC6595/6A595 ≙ 595 - 8-Bit-Schieberegister mit Halteregister. Kaskadierbar..

Treiber mit Rücklesemöglichkeit, Strombegrenzung usw.
High Side Drive.
Halbbrücken- und Brückenschaltungen
Treiber für Leistungs-FETs.

Auf Last achten. Induktive Lasten. Beschaltung mit Freilaufdioden - so, daß Stromweg geschlossen wird. Auf Sperrträgheit achten, bes. bei Nutzung der parasitären Dioden in Leistungs-FETs.
Leistungs-FETs schlagen über das Gate zurück.

Zum universellen Bus-Transceiver 16600
Ein Schaltkreis Typs enthält 18 Bitpositionen. Die Steuereingänge sind allen Bitpositionen gemeinsam. Wir betrachten im folgenden nur die Signalflußrichtung von links nach rechts bzw. von A nach B. Um die gewünschte Richtung einzustellen, sind /OEAB mit Low und /OEBA mit High zu belegen. Flipflop 2 (Gegenrichtung) wird zum Zurücklesen ausgenutzt. Im Normalbetrieb sind die zugehörigen Steuereingänge inaktiv geschaltet. Zur Datenausgabe wird ausschließlich den Takteingang CLKAB verwendet. Die Betriebsweise der Flipflops wird mit LEAB gesteuert:

- LEAB = Low: Flipflop 1 übernimmt mit der Low-High-Flanke des Taktes CLKAB die Belegung des A-Eingangs (Registerfunktion). Damit CLKAB wirksam werden kann, muß /CLKENAB = Low sein.
- LEAB = High: Flipflop 1 wirkt nicht als Speicherglied, sondern als Durchgang (Schaltkreis hat lediglich Treiber-Funktion).

Latch-Betrieb:
Solange LEAB = High ist, wirkt Flipflop 1 als Durchgang. Wird LEAB auf Low geschaltet, so wirkt Flipflop 1 als Speicherglied und hält ausgangsseitig die Eingangsbelegung zum Zeitpunkt der High-Low-Flanke von LEAB (Verhalten eines transparenten Latches). Soll Flipflop 1 als Latch betrieben werden, darf CLKAB bei LEAB = Low nicht wirksam werden (sonst: Informationsübernahme mit Low-High-Flanke).

Zurücklesen:
Über LEBA ist wählbar, ob Flipflop 2 als Durchreiche (LEBA = High) oder als Speicherelement*) verwendet werden soll. Zum eigentlichen Zurücklesen: (1) Datenport des Mikrocontrollers auf Eingabe schalten, (2) /OEBA aktivieren, (3) Daten lesen, (4) /OEBA deaktivieren, (5) Datenport wieder auf Ausgabe schalten.

*): Schaltverhalten entweder als Latch (Speichern der Belegung mit High-Low-Flanke von LEBA; CLKBA ständig inaktiv) oder als D-Flipflop (LEBA fest auf Low, Übernahme mit Low-High-Flanke von CLKBA).

Hinweise:

1. CLK-Signale = Takte für D-Flipflops,
2. LE-Signale = Takte für Latches.
3. Es gibt verschiedene Typen ähnlicher Schaltkreise. Sie unterscheiden sich in der Breite und in Einzelheiten der Ansteuerung (so gibt es Typen, die - bei ansonsten gleicher Funktionsweise - keine Takterlaubniseingänge (/CLKENAB, /CLKENBA) haben (16500/501).

Nutzung des Schieberegister- bzw. Scan-Prinzips

Der Vorteil des Schieberegisters: mit nur wenigen Anschlüssen am Mikrocontroller (typischerweise 3 oder 4) können beliebig viele Bitpositionen als Ein- oder Ausgänge vorgesehen werden. Um dieses Prinzip auf kostengünstige Weise auszunutzen, bieten sich folgende Möglichkeiten an:

- der Einsatz von Treiberstufen mit Boundary-Scan-Vorkehrungen (nach JTAG/IEEE1149.1),
- der Einbau entsprechender Schiebewege in CPLDs bzw. FPGAs.

Industriestandards:

- JTAG 1149.1,
- I^2C-Bus (Philips),
- SPI (Motorola),
- Microwire (NSC).

Praxistips zur Aufwandsverringerung:

1. Schiebetakte,Übernahme-Strobes usw. direkt vom Mikrocontroller aus über Software steuern (einfacher als durchlaufender Takt + JTAG-ähnliche State Machine).
2. Daß sich das Durchschieben an den Schaltkreisanschlüssen bemerkbar macht, stört oftmals nicht. Dann braucht man auch kein Halteregister zwischenzuschalten (Ersparnis von Parallelregister-Flipflops).
3. Halteregister können (in FPGAs) mit Latches aufgebaut werden (spart Zellen).
4. Zur Abfrage von Signalen ist oftmals das Serializerprinzip (Abfrage über Multiplexer) günstiger: (1) kein Flipflop-Verbrauch (CPLDs), (2) Zellen sind ohnehin für Multiplexer-Strukturen optimiert (manche FPGAs).

Signalabfrage

- Serializerprinzip
- Matrixorganisation
- Schiebeprinzip

Einfachinterfaces

Aufwands-Größenordnung

Einzelleitungen: O(n)

Mux/Demux: O(ld n)

Schieberegister oder echter Bus: const.

*): (mit Adreßdecodierung in den Einrichtungen

Kompromißlösungen mit zentraler Auswahl (SPI, Microwire usw.) haben O(n) (Einzelauswahl)
oder O(ld n) (mit externer Adreßdecodierung).

Industriestandards für einfache serielle Interfaces

I2C-Bus (Philips)

- 2 Leitungen
- Open Drain
- echter Bus,
- keine Contention,
- Adreßdecodierung in Einrichtungen,
- echtes Multimaster-Protokoll,
- massiv standardisiert

Microwire (Natsemi)

- 4 Leitungen
- Tristate-Ausgänge,
- zentraler Master,
- Bus mit Einzelauswahl der Slave-Einrichtungen (keine Schiebekette),
- Taktpolarität und Schiebe-Ordnung festliegend,
- mit Ausnahme von Industriestandard-Einrichtungen (z. B. seriellen EEPROMs) weitgehende Narrenfreiheit

SPI (Synchronous Peripheral Interconnect; Motorola)

- 4 Leitungen
- Tristate-Ausgänge,
- zentraler Master,
- Bus mit Einzelauswahl der Slave-Einrichtungen (keine Schiebekette),

- Taktpolarität und Schiebeordnung programmierbar,
- mit Ausnahme von Industriestandard-Einrichtungen (z. B. seriellen EEPROMs) weitgehende Narrenfreiheit

Zurücklesen von Ausgaben

a) um den Maschinenzustand zu kennen und um modifizierende Befehle auf die ausgegebenen Daten anwenden zu können,

b) zum Abholen des tatsächlichen Zustandes,

c) zu Prüfzwecken (Stimulus - Response).

Prinzipien:

Verwalten mittels Software
Ausgaben nur über Macros bzw. Unterprogramme, die entsprechende Kopien im RAM oder in Registern halten. Nützt nichts zu den Zwecken b), c).

Hardwareseitiges Mitführen von RAM-Kopien
Parallel zur Ausgabe wird RAM adressiert und in denselben geschrieben. Funktioniert nur bei externem Speicherbus (8051 usw.). Nützt nichts zu den Zwecken b), c).
Lösungen

Zurücklesen aus Hardware
Bei den typischen (moderneren) Mikrocontrollern möglich. Genau nachsehen, welche Befehle was wirklich lesen (Register oder Pin?) - z. B. bei 8051.

Rücklesbare Latches (Readback Latches; z. B. von TI)
Exoten. Nur 5 V. Verfügbarkeit beachten.

Universelle Bus-Transceiver
Ermöglichen Schalten von Rücklesewegen. Vorzugslösung.

Schiebewege
Rückführung auf Schieberegister mit parallelen Eingängen.

Multiplexer
Können auch weiter hinten ansetzen (Abfrage beliebiger Signale aus der umgebenden Schaltung - auch von Analogsignalen (Stichwirte: Analog-MUXe, Comparatoren, A/D-Wandler).

Prüfen bidirektionaler Signalwege
Nutzung der parasitären Kapazitäten als Ladungsspeicher: Ausgeben und sofort wieder zurücklesen.

Adreßverlängerung

Heutzutage nur Einfachlösungen. das typische Szenarium: Mikrocontroller genügt an sich, es ist lediglich ein größerer Datenspeicher erforderlich (z. B. SRAM oder Flash von mehreren MBytes).

1. Nutzung von weiteren E-A-Ports, um die höchstwertigen Adreßbits zu liefern

2. Bankregister außen. Laden bei Zugriff auf bestimmte Adressen (Memory Mapped I/O). Memory Mapper-Schaltkreise (auch als Entwurfs-Anregung (CPLD)).

3. Bankregister draußen. 1 zusätzliche Port-Leitung. Um Register zu laden, ein blindes Lesen auslösen => Adreßbelegung wird in Register übernommen. Auf Nebenwirkungen des blinden Lesens achten (Flash-Programmierung??). Ermöglicht auch ganz allgemein breite Ausgaben (z. B. zu 16 Bits).

4. Befehlsspeicher (1). Segmentieren: Banked/Unbanked

5. Befehlsspeicher (2). Eine Möglichkeit: Adreßraum nicht erweitern, aber teilweise mit RAM bestücken. Programme transient laden.

6. Befehlsspeicher (3). Eigene (bessere) Befehlsliste festlegen und Interpreter schreiben. Erlaubt, praktisch alle Einschränkungen der Befehlsliste des µC zu übergehen.