TECHNISCHE UNIVERSITÄT DRESDEN

Fakultät Informatik, Institut für Technische Informatik, Professur für VLSI-Entwurfssysteme, Diagnostik und Architektur

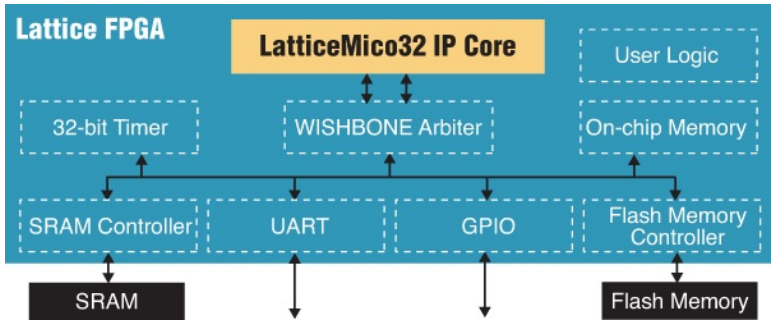
WISHBONE SoC Interconnection

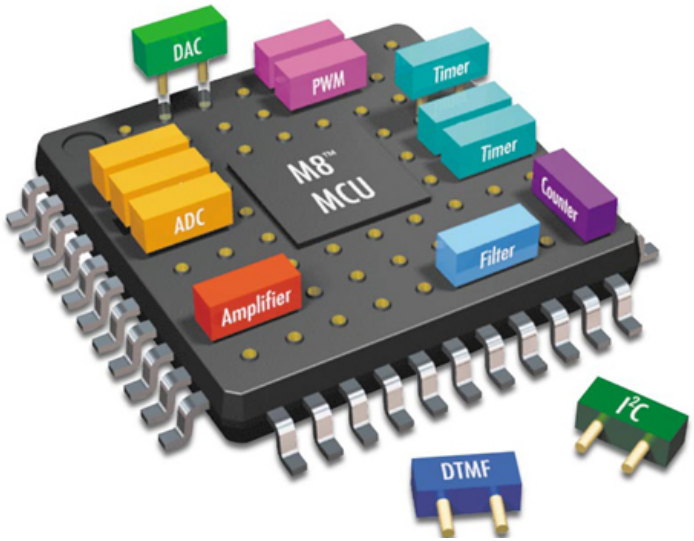
Einführung und Überblick

Carsten Herrmann

Dresden, 13.06.12

Motivation





Wünsche:

- flexible Architektur
- leichte Wiederverwendung durch kompatible Schnittstellen
- strukturierte Entwurfsmethoden

www.opencores.org
„the #1 community within open source hardware IP-cores“

http://www.latticesemi.com/images/img40901.jpg
http://www.elv-downloads.de/bilder/journal/2004_06/16/2004_06_16_psoc_kopfbild.jpg
http://www.cypress.com/psoc

Gliederung

1. Einführung
2. WISHBONE-Interface
   - Signale
   - Übertragungsprotokoll
     - Handshaking (Standard Mode)
     - Classic Cycles
     - Registered Feedback Cycles
3. Integration
   - Freiheitsgrade und WISHBONE Datasheet
   - INTERCON
4. Rechtliche Aspekte
5. Anhang: Pipelined Mode

Grundlagen

Entwurfsziele und Eigenschaften

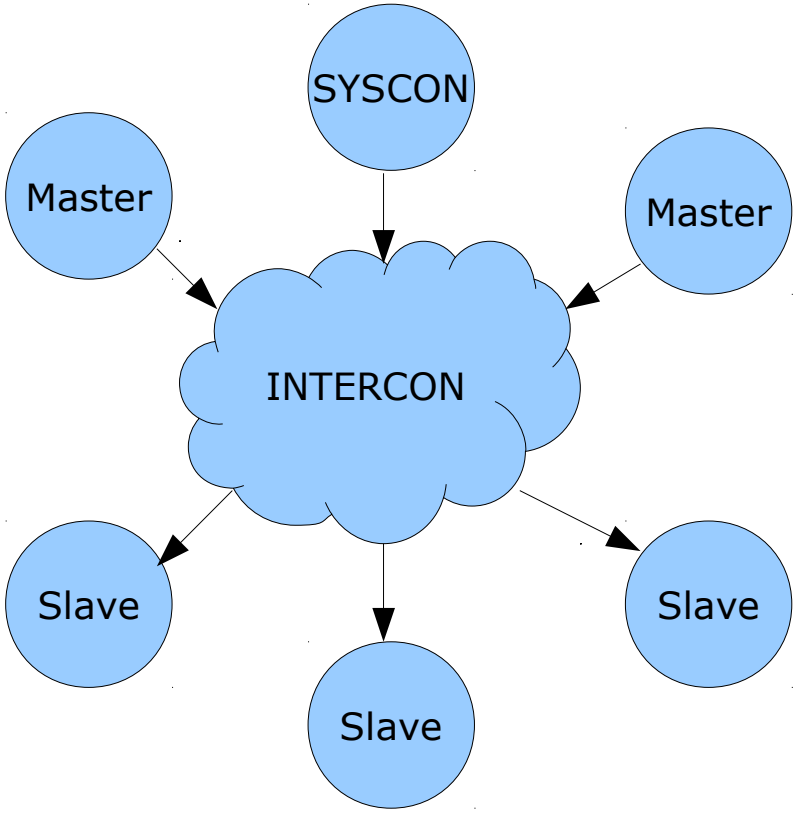
- einfach, kompakt, wenig Glue Logic
- variable Busbreiten (Address: 0...64, Data: 8, 16, 32, 64) und Byteorder
- Unterstützung aller gängigen Transferzyklen inkl. Read/Write, Block Transfer und Read-Modify-Write
- Master-Slave-Architektur mit Multi-Master-/Slave-Unterstützung
- variables Verbindungsnetzwerk mit Unterstützung für Punkt-zu-Punkt-, Shared Bus- und Switched Fabric-Topologien; festgelegt vom Systemintegrator
- synchrones Design, einfache Timing-Spezifikation
- Unterstützung für Single-Clock-Transfers und Übertragungsratendrosselung durch Handshaking-Protokoll
- Unterstützung von nutzerspezifischen Address-, Data- und Cycle-Tags
- unabhängig von Technologie, Toolchain (auch HDL) und Vertriebsmethode
- minimaler Dokumentationsstandard (-> WISHBONE Datasheet)

[1], S. 9ff.

Grundlagen

Ansatz

- Master-Slave-Architektur
- standardisierte Schnittstelle(n)
- variables Verbindungsnetzwerk
  - Point-to-Point
  - Shared Bus
  - Crossbar Switch
  - Data Flow
  - ...



[1], S. 92

Gliederung

Signale

System (SYSCON)

CLK (Clock):
- globales Taktsignal

RST (Reset):
- globales Reset-Signal
- synchron zu CLK

➔ synchrones Design, alle WISHBONE-Signale bezogen auf CLK (steigende Flanke) (Ausnahme: asynchrone Cycle Termination im Classic Mode (s.u.))

➔ CLK und RST beziehen sich nur auf WISHBONE-Interfaces
- andere Teile eines IP-Cores müssen nicht angebunden sein (dürfen aber)
- es kann weitere (z.B. auch asynchrone) Reset-Signale geben

[1], 2.2.1 S. 27

Signale

Handshake

Master

CYC (Cycle):
- zeigt aktiven Bus Cycle an
- erklärt andere Signale als gültig

STB (Strobe):
- markiert Transferanforderung
- erklärt zugehörige Signale als gültig

optional:

LOCK (Lock):
- verriegelt Arbitrierung

Slave

ACK (Acknowledge):
- markiert normales Transferende
- erklärt ggf. Lesedaten als gültig

optional:

ERR (Error), **RTY** (Retry):
- Signalisierung anstatt ACK
- zeigt an, dass Transfer nicht korrekt ausgeführt wurde (ERR) oder wiederholt werden soll (RTY)

STALL (Stall):
- für Pipelined Mode
- markiert Pipeline-Blockierungen

[1], 2.2, S. 28-30

Signale

Transferdaten

Master

ADR (Address)

DAT_O (Data):
Schreibdaten (bei Schreibzugriff)

DAT_I (Data):
Lesedaten (bei Lesezugriff)

WE (Write Enable):
markiert Zugriffsrichtung (Lesen oder Schreiben)

SEL (Byte Select):
markiert gültige Bereiche in DAT

Slave

DAT_I (Data):
Schreibdaten (bei Schreibzugriff)

DAT_O (Data):
Lesedaten (bei Lesezugriff)

[1], 2.2, S. 27-30

Signale

Tags

Master

TGD (Cycle Tag):
- Tags zum gesamten Cycle
- Timing wie CYC
- Bsp.: SGL, BLK, RMW

TGA (Address Tag):
- Tags zur aktuellen Adresse
- Timing wie ADR
- Bsp.: CTI, BTE

TGD (Data Tag):
- Tags zu aktuellen Schreibdaten
- Timing wie DAT

Slave

TGD (Data Tag):
- Tags zu aktuellen Lesedaten
- Timing wie DAT

[1], 2.2, S. 27-29

Signale

Namenskonventionen

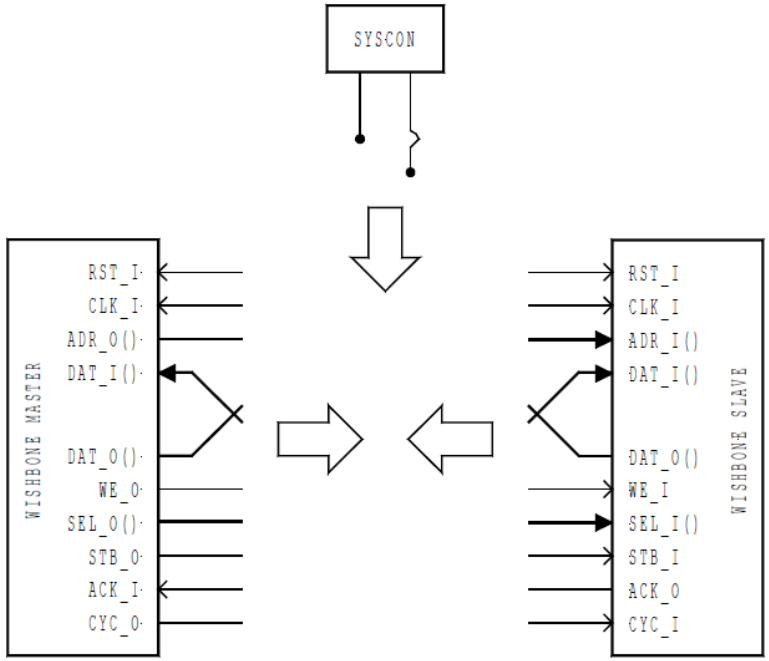
- Empfehlung: Signale entsprechend WISHBONE-Spezifikation benennen
- Verwendung anderer Namen ist erlaubt, z.B. zur Anpassung an Namenskonventionen der HDL oder bei Cores mit mehreren WISHBONE-Interfaces
- für jedes Signal mit abweichendem Namen muss Querverweis auf zugehörigen Signalnamen entsprechend Spezifikation im WISHBONE-Datasheet angegeben werden

- zur Kennzeichnung der Datenrichtung werden Suffixe _I (Input) und _O (Output) verwendet
- in Taktdiagrammen werden Signale aus Sicht des Masters dargestellt

[1], 2.1.2, S. 26

Signale

Punkt-zu-Punkt-Verbindung



(A) FORMING A POINT-TO-POINT INTERCONNECTION.

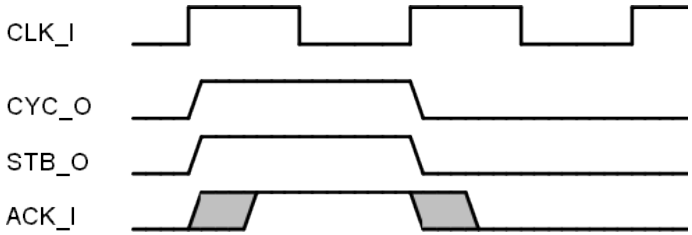
[1], Abb. 8-6 (A), S. 97

Gliederung

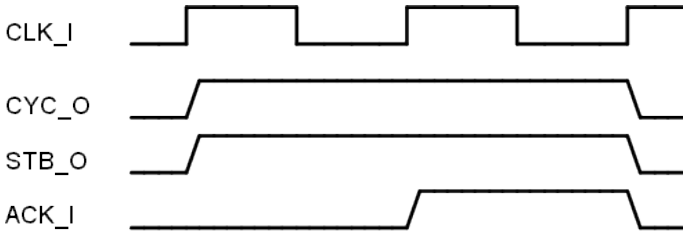
Übertragungsprotokoll (Standard Mode)

Grundlegendes Handshaking

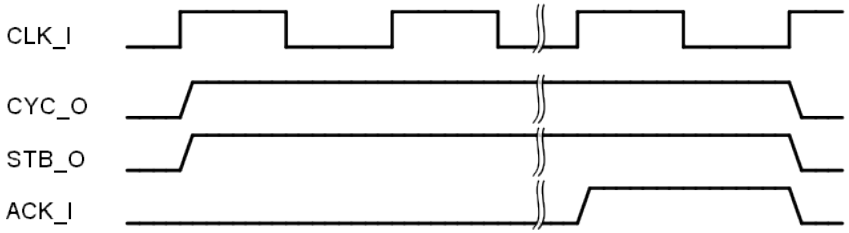
asynchron



synchron



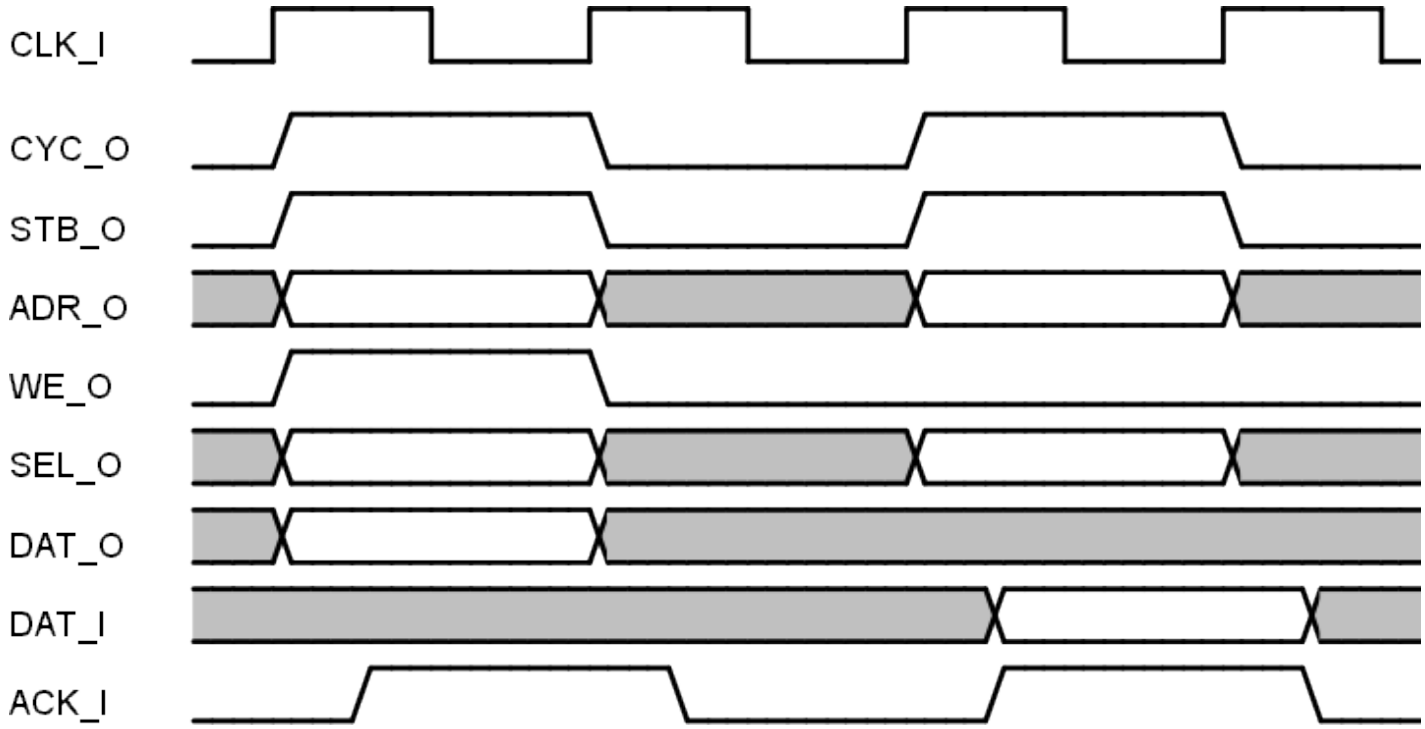
mit Waitstates



[1], 3.1.2 - 3.1.3, S. 33-37

Übertragungsprotokoll (Standard Mode)

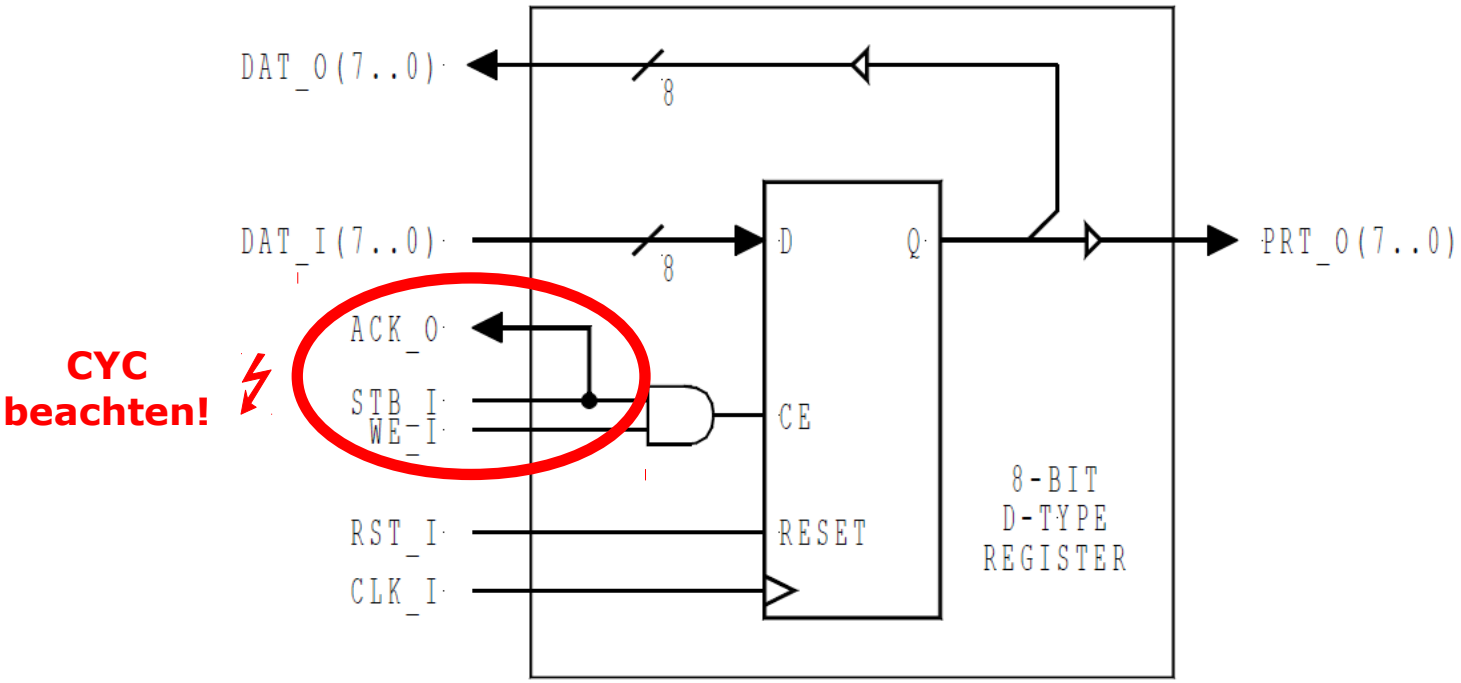
Single Read/Write, asynchrones ACK



[1], 3.2, S. 39ff.

Übertragungsprotokoll (Standard Mode)

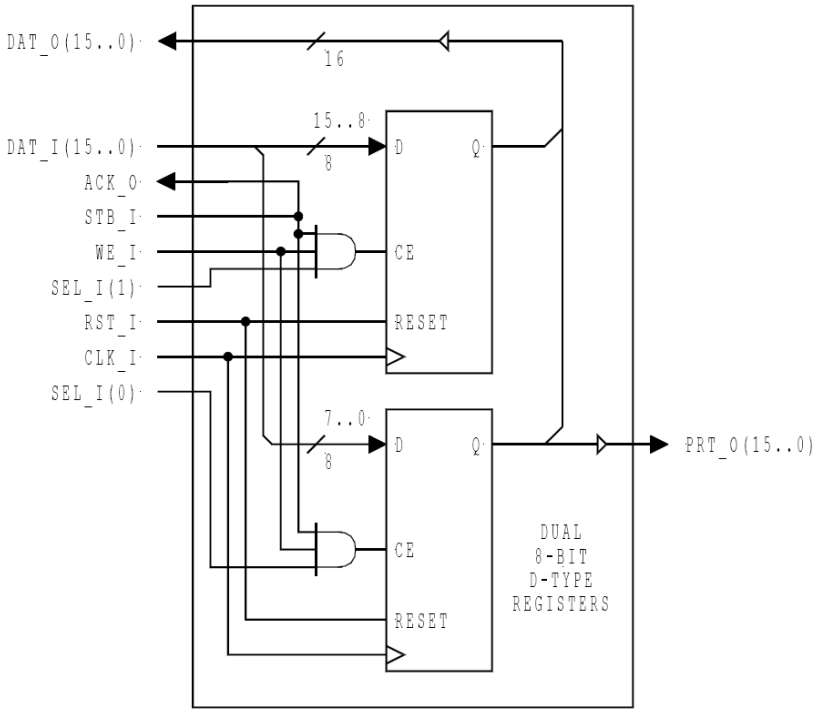
Beispiel: 8-Bit Output-Port



[1], Abb. 8-9, S. 102

Übertragungsprotokoll (Standard Mode)

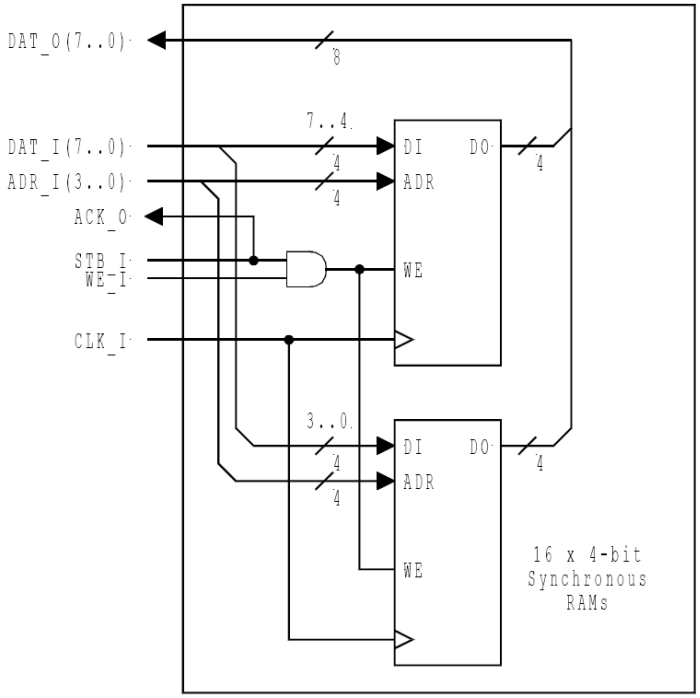
Beispiel: 16-Bit Output-Port mit 8-Bit Granularität



[1], Abb. 8-11, S. 106

Übertragungsprotokoll (Standard Mode)

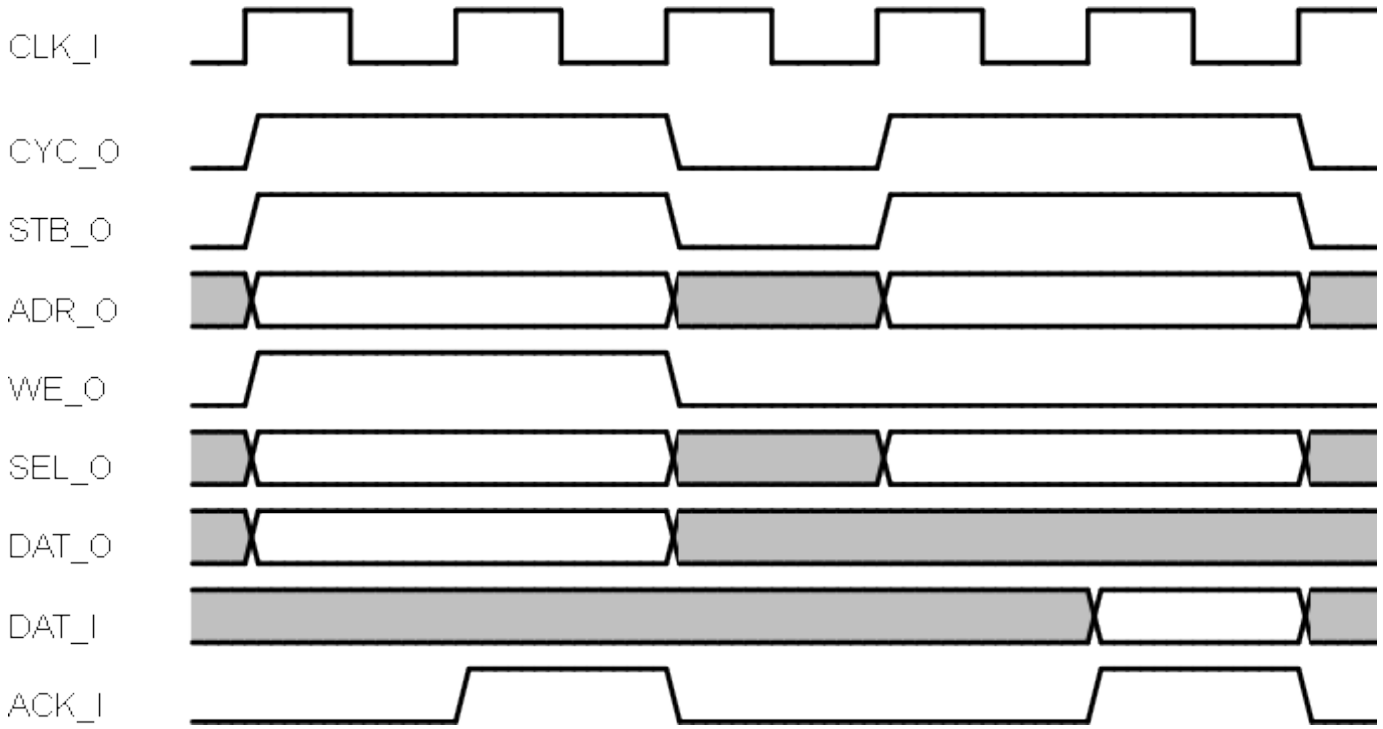
Beispiel: 16x8-Bit RAM



[1], Abb. 8-14, S. 111

Übertragungsprotokoll (Standard Mode)

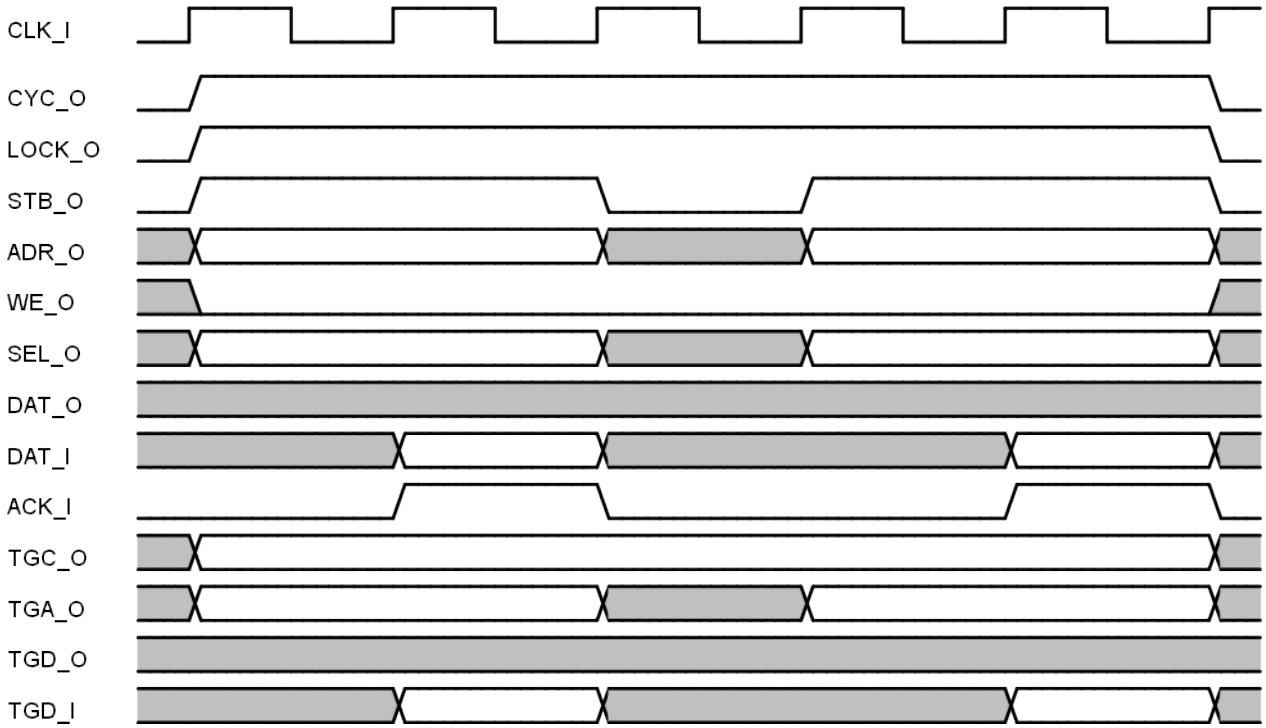
Single Read/Write, synchrones ACK



[1], 3.2, S. 39ff.

Übertragungsprotokoll (Standard Mode)

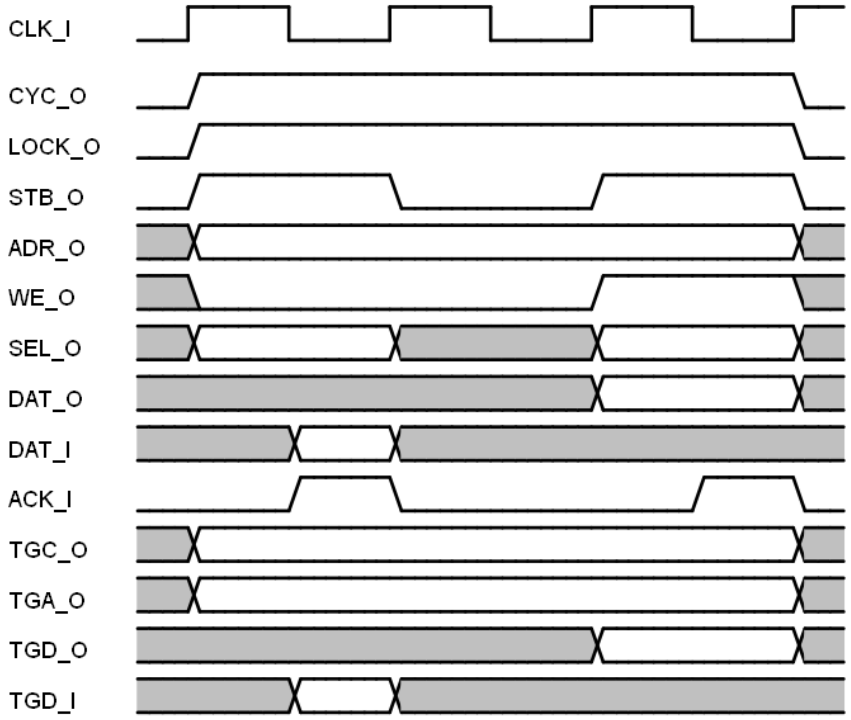
Block Read



[1], 3.3, S. 47ff.

Übertragungsprotokoll (Standard Mode)

Read-Modify-Write

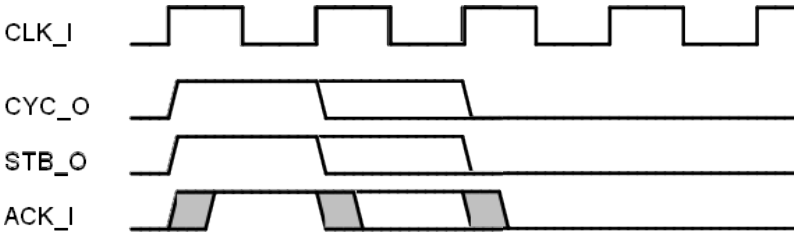


[1], 3.4, S. 56f.

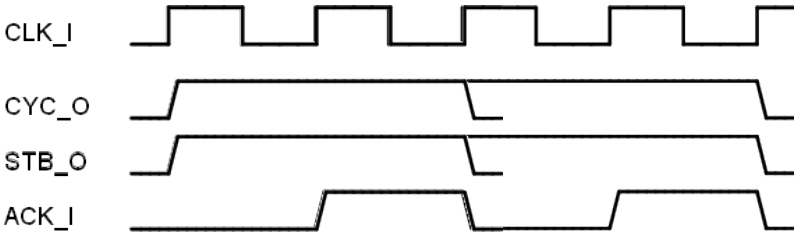
Übertragungsprotokoll (Standard Mode)

Registered Feedback Cycles

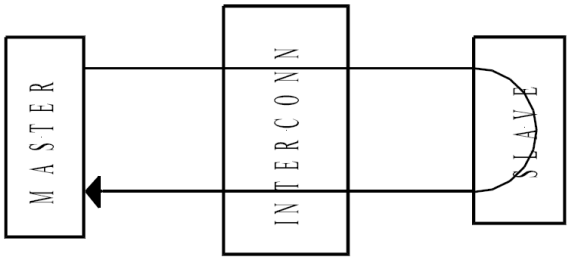
asynchron



synchron



asynchrone Rückführung

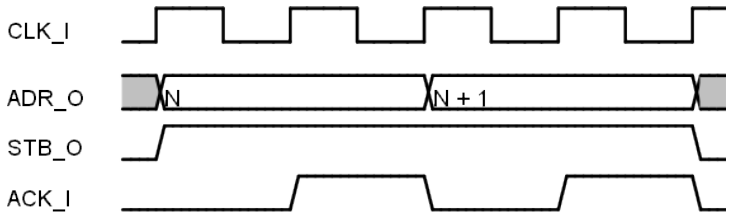


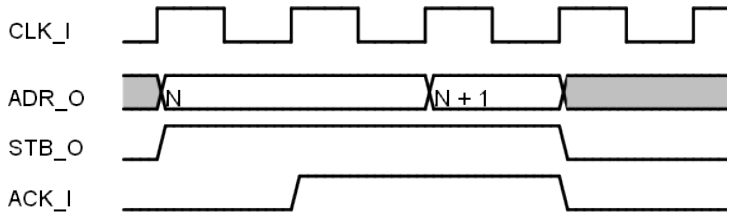
2 Takte pro Transfer =
50% Performance-Verlust

[1], 4.1, S. 66ff.

Übertragungsprotokoll (Standard Mode)

Registered Feedback Cycles





Burst-Länge	#Takte asynchron	#Takte synchron	#Takte erweitert
1	1 (200%)	2 (100%)	2
2	2 (150%)	4 (75%)	3
4	4 (125%)	8 (62%)	5
8	8 (112%)	16 (56%)	9
16	16 (106%)	32 (53%)	17

[1], 4.1, S. 66ff.

Übertragungsprotokoll (Standard Mode)

Registered Feedback Cycles

Einführung zusätzlicher Address-Tags zur Kennzeichnung des Cycle Types:

CTI (Cycle Type Identifier):

000	Classic Cycle
001	Constant Address Burst
010	Incrementing Burst
...	reserviert
111	End of Burst

BTE (Burst Type Extension):

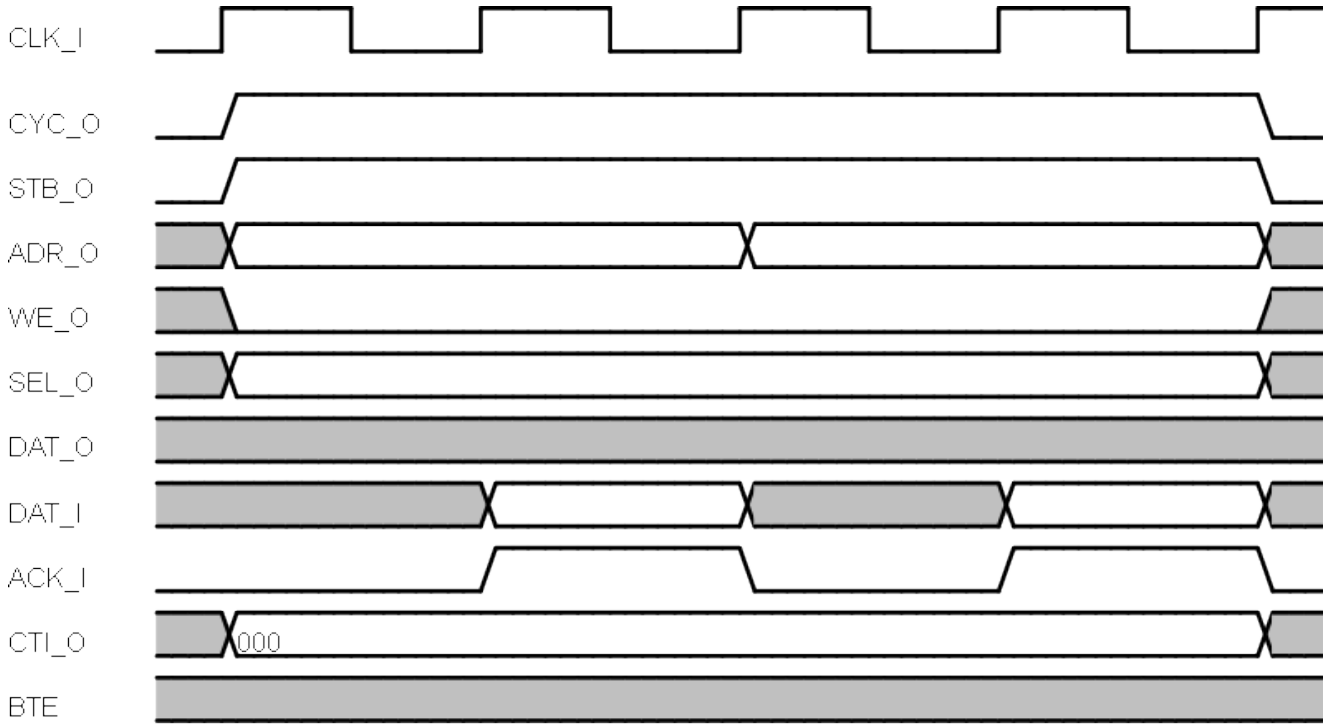
00	Linear Burst
01	4-Beat Wrap Burst
10	8-Beat Wrap Burst
11	16-Beat Wrap Burst

➔ Slave kann nächsten Zugriff „vorhersehen" und schon vorbereiten (darf dazu ACK/ERR/RTY unabhängig von STB generieren)

[1], 4.3, S. 69ff.

Übertragungsprotokoll (Standard Mode)

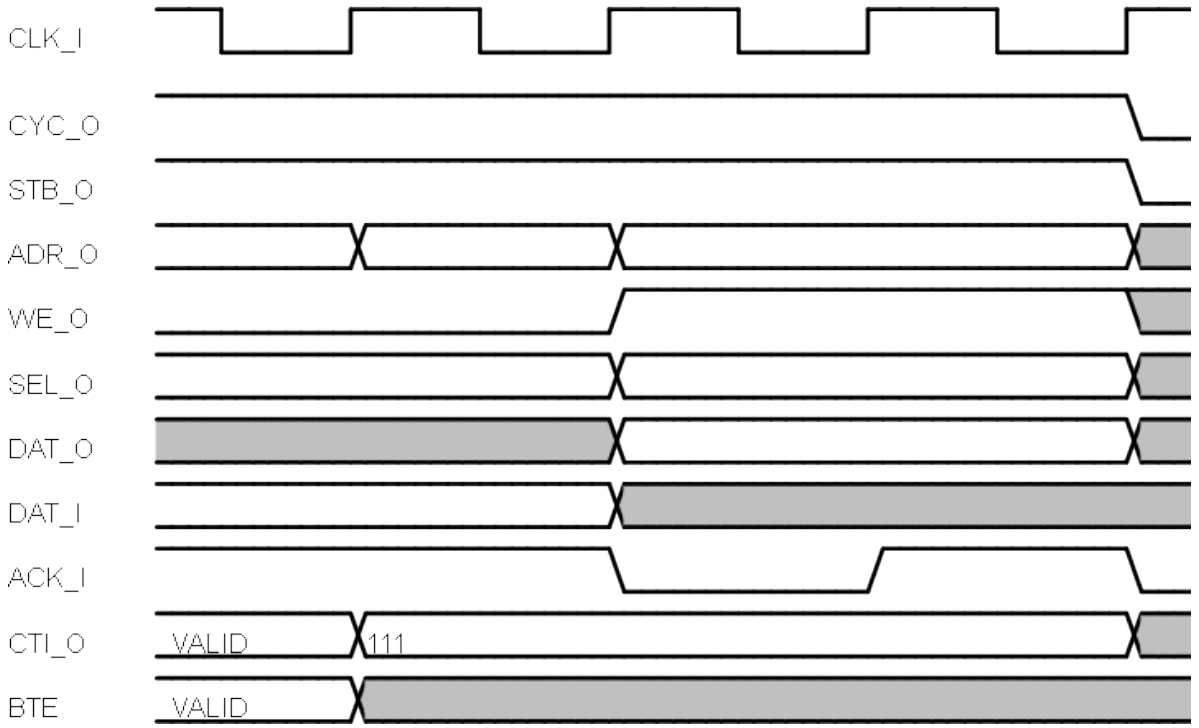
Registered Feedback - Classic Cycle



[1], 4.4.1, S. 71ff.

Übertragungsprotokoll (Standard Mode)

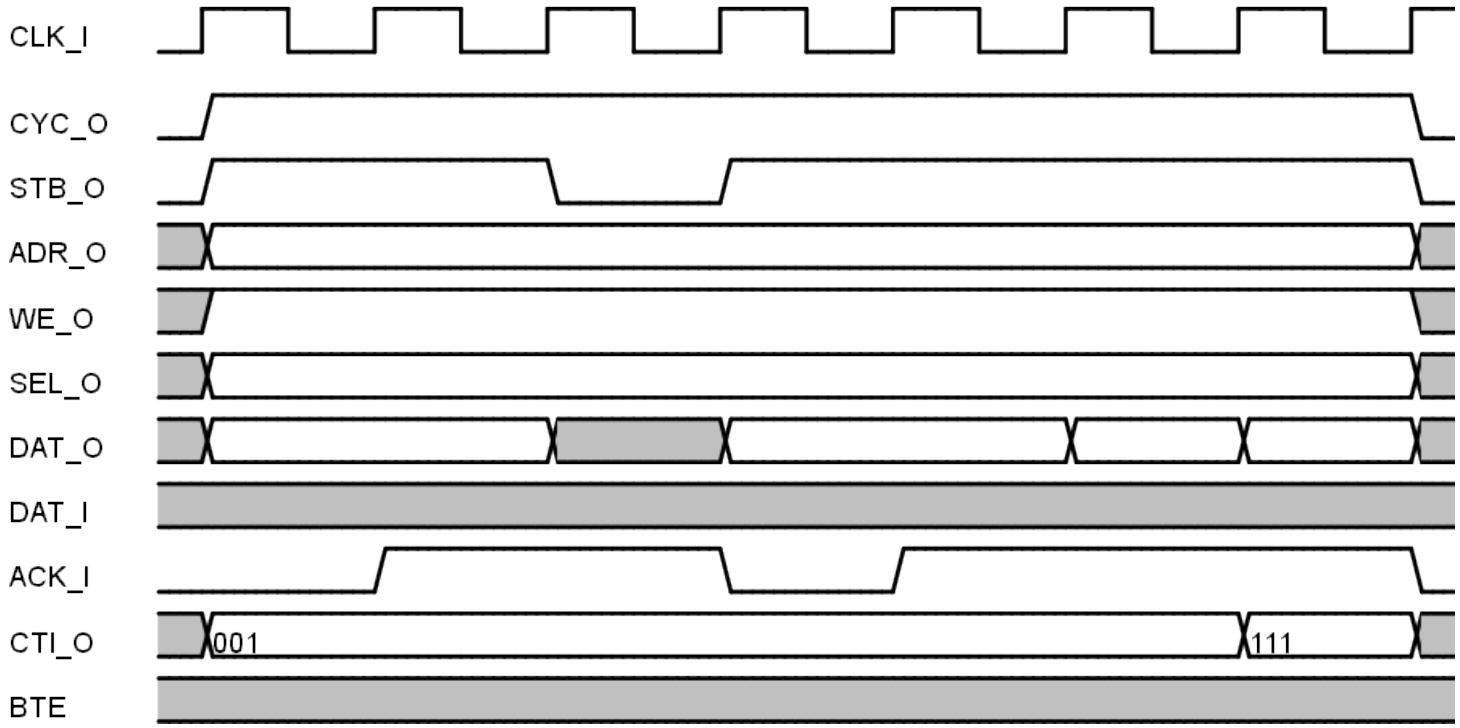
Registered Feedback - End-of-Burst



[1], 4.4.2, S. 73ff.

Übertragungsprotokoll (Standard Mode)

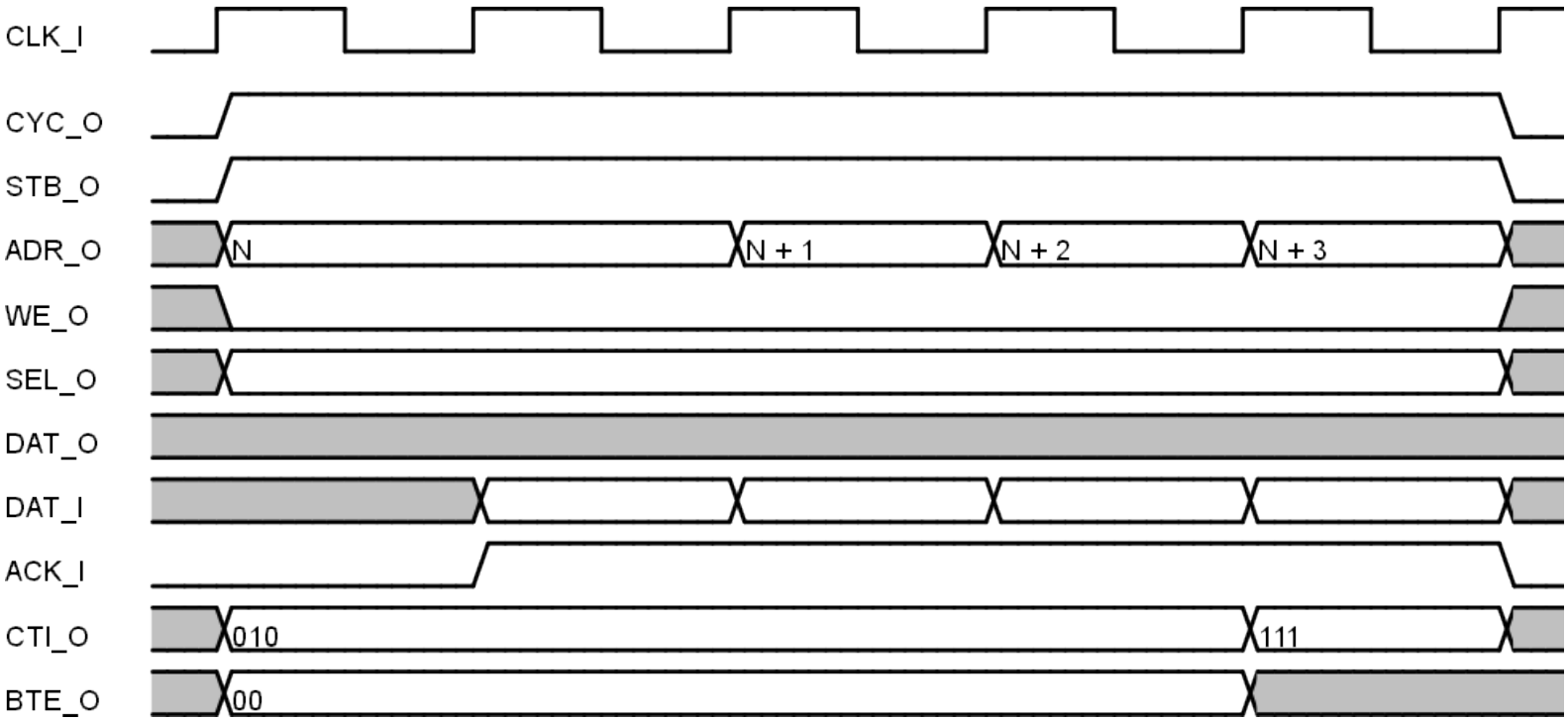
Registered Feedback – Constant Address Burst



[1], 4.4.3, S. 76ff.

Übertragungsprotokoll (Standard Mode)

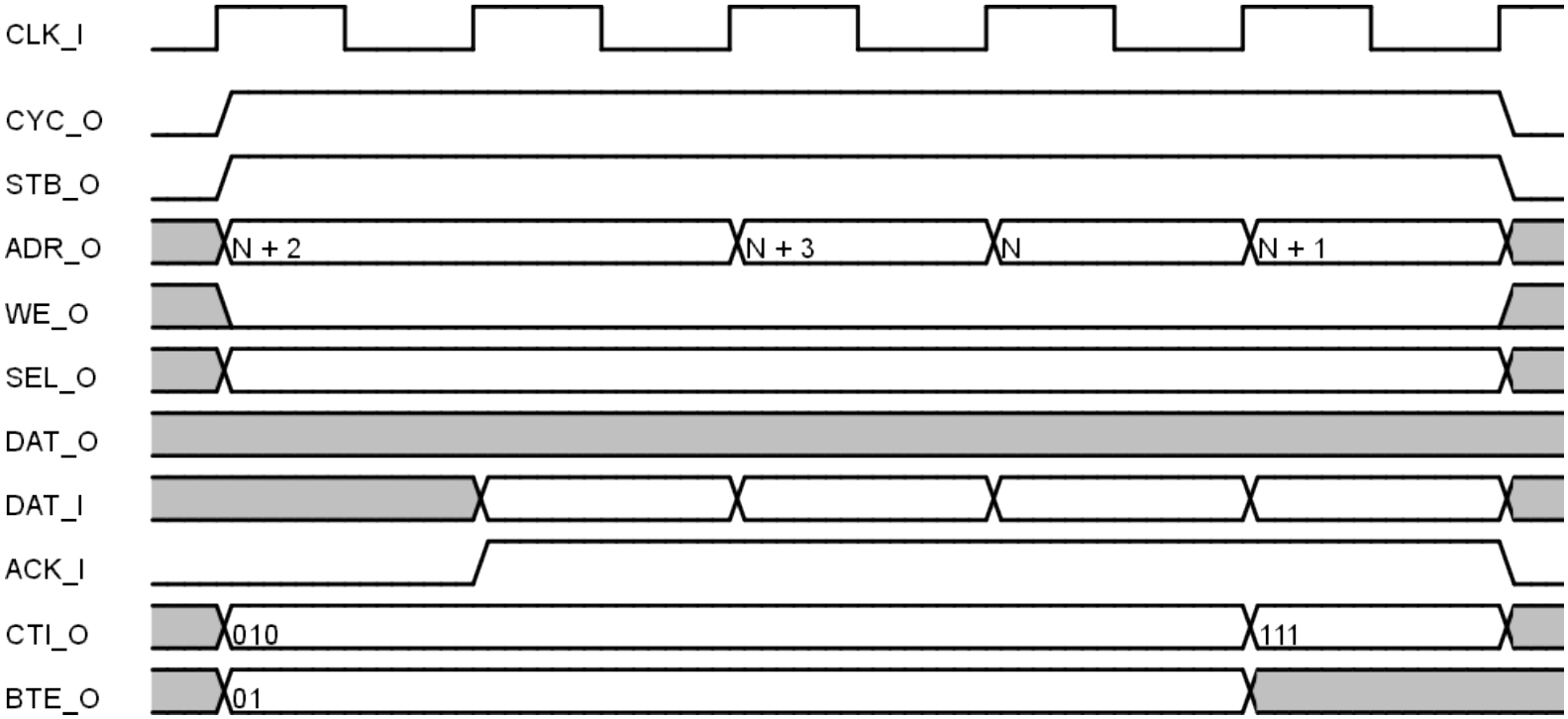
Registered Feedback – Incrementing Burst



[1], 4.4.4, S. 79ff.

Übertragungsprotokoll (Standard Mode)

Registered Feedback – 4-Beat Wrap Burst



[1], 4.4.1, S. 79ff.

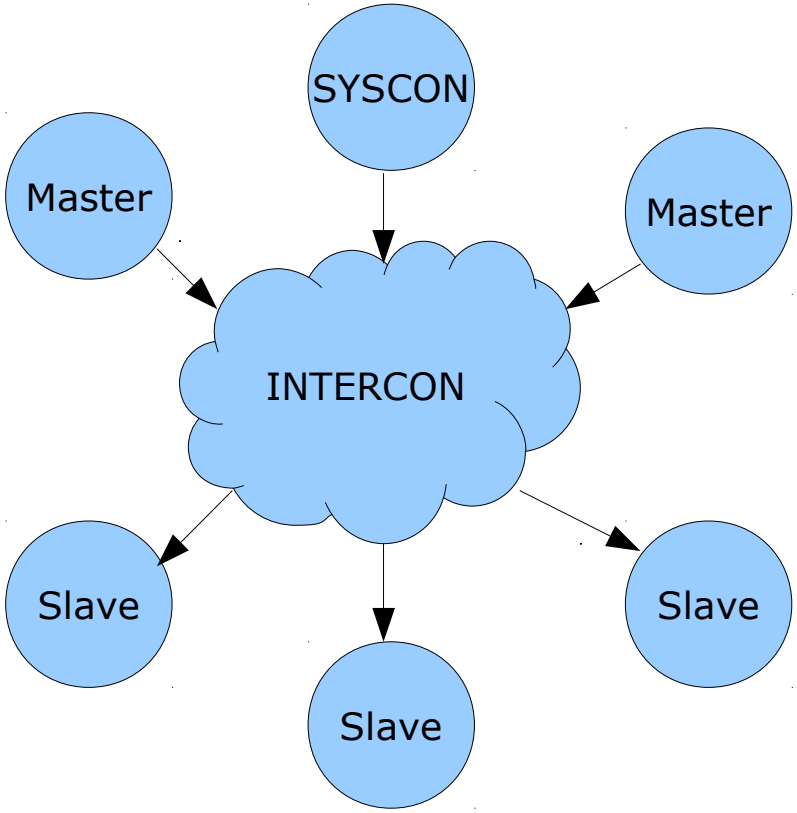
Gliederung

1. Einführung
2. WISHBONE-Interface
   - Signale
   - Übertragungsprotokoll
     - Handshaking (Standard Mode)
     - Classic Cycles
     - Registered Feedback Cycles
3. Integration
   - Freiheitsgrade und WISHBONE Datasheet
   - INTERCON
4. Rechtliche Aspekte
5. Anhang: Pipelined Mode

Integration

Ansatz

- Ausgangspunkt: IP-Cores mit standardisiertem Interface
  - vereinfachen strukturierten Entwurf und Integration
- Systemintegrator realisiert Elemente der Verbindungs-architektur („Variable Interconnection“)
  - SYSCON
  - INTERCON



[1], Abb. 8-1, S. 92

Integration

WISHBONE-kompatible Cores

- standardisiertes Interface mit definierten Freiheitsgraden
  - Signale und Protokoll (optionale Signale, unterstützte Cycle Types, ...)
  - Data Organization (Busbreiten, Granularität, Byteorder, ...)
  - Tags (TAGTYPE, Bedeutung, ...)
- WISHBONE Datasheet
- Logo

Integration

WISHBONE-kompatible Cores

Table A-1. WISHBONE DATASHEET for the 8-bit output port example.	
Description	Specification
General description:	8-bit SLAVE output port.
Supported cycles:	SLAVE, READ/WRITE SLAVE, BLOCK READ/WRITE SLAVE, RMW
Data port, size: Data port, granularity: Data port, maximum operand size: Data transfer ordering: Data transfer sequencing:	8-bit 8-bit 8-bit Big endian and/or little endian Undefined
Supported signal list and cross reference to equivalent WISHBONE signals:	Signal Name — WISHBONE Equiv. ACK_O — ACK_O CLK_I — CLK_I DAT_I(7..0) — DAT_I() DAT_O(7..0) — DAT_O() RST_I — RST_I STB_I — STB_I WE_I — WE_I

[1], Tab. A-1, S. 103

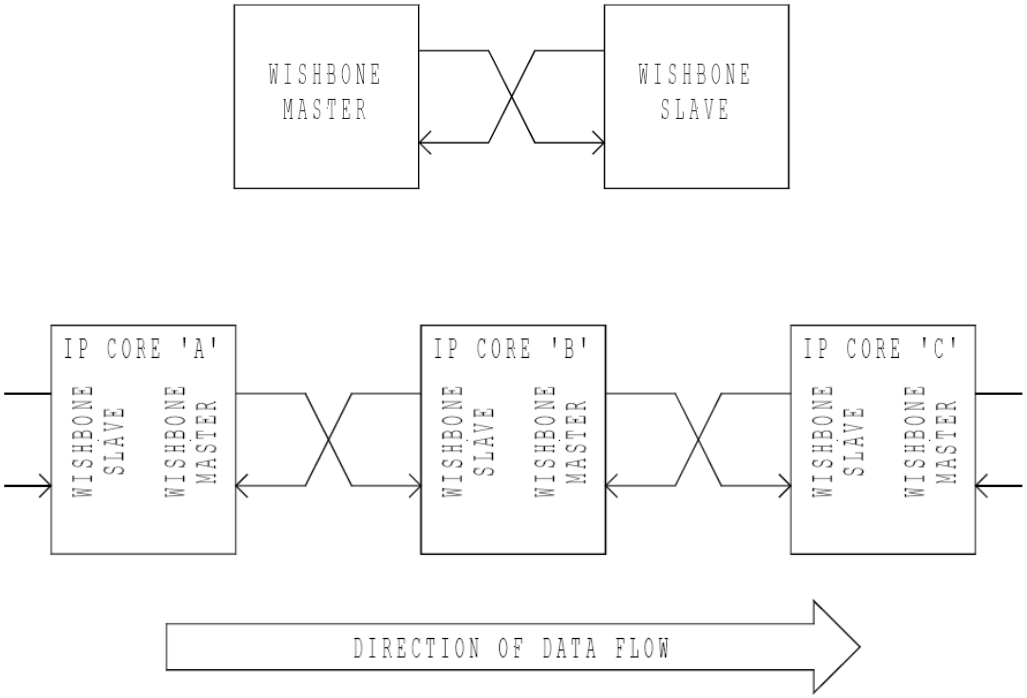
Integration

INTERCON

- Verwendung/Abwandlung bestehende Implementierung oder eigene Realisierung
- wesentliche Schwerpunkte:
  - Topologie
  - Adressdekodierung
  - Arbitrierung

Integration

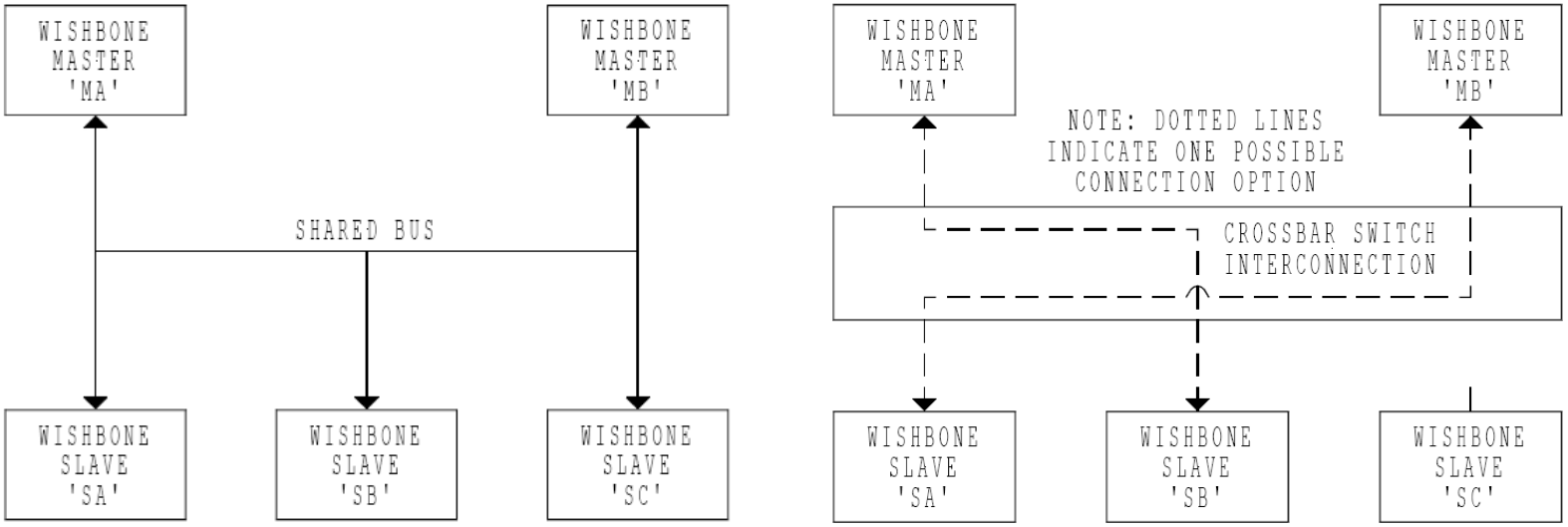
INTERCON - Topologien



[1], 8.2, S. 93ff.

Integration

INTERCON - Topologien

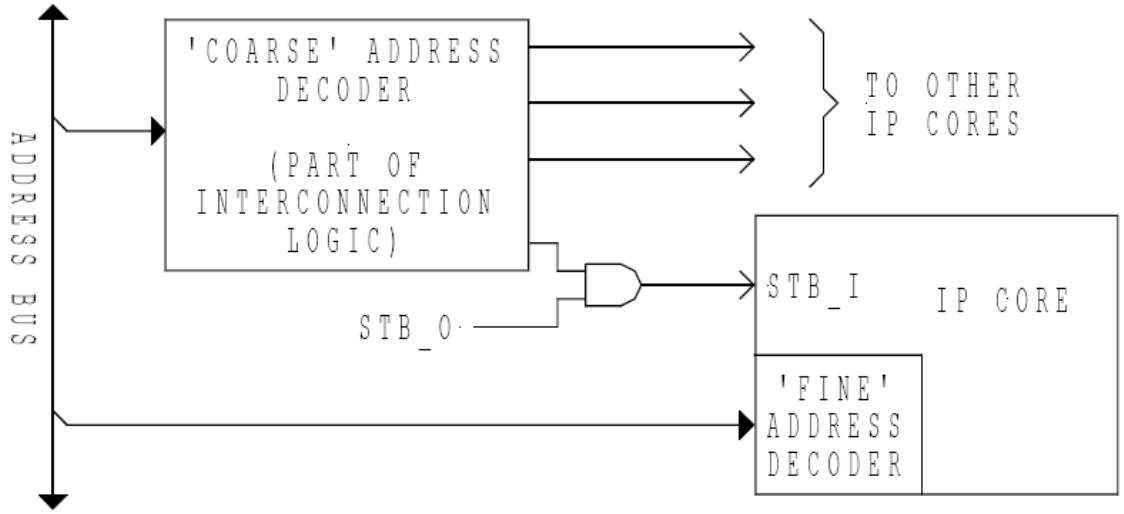


[1], 8.2, S. 93ff.

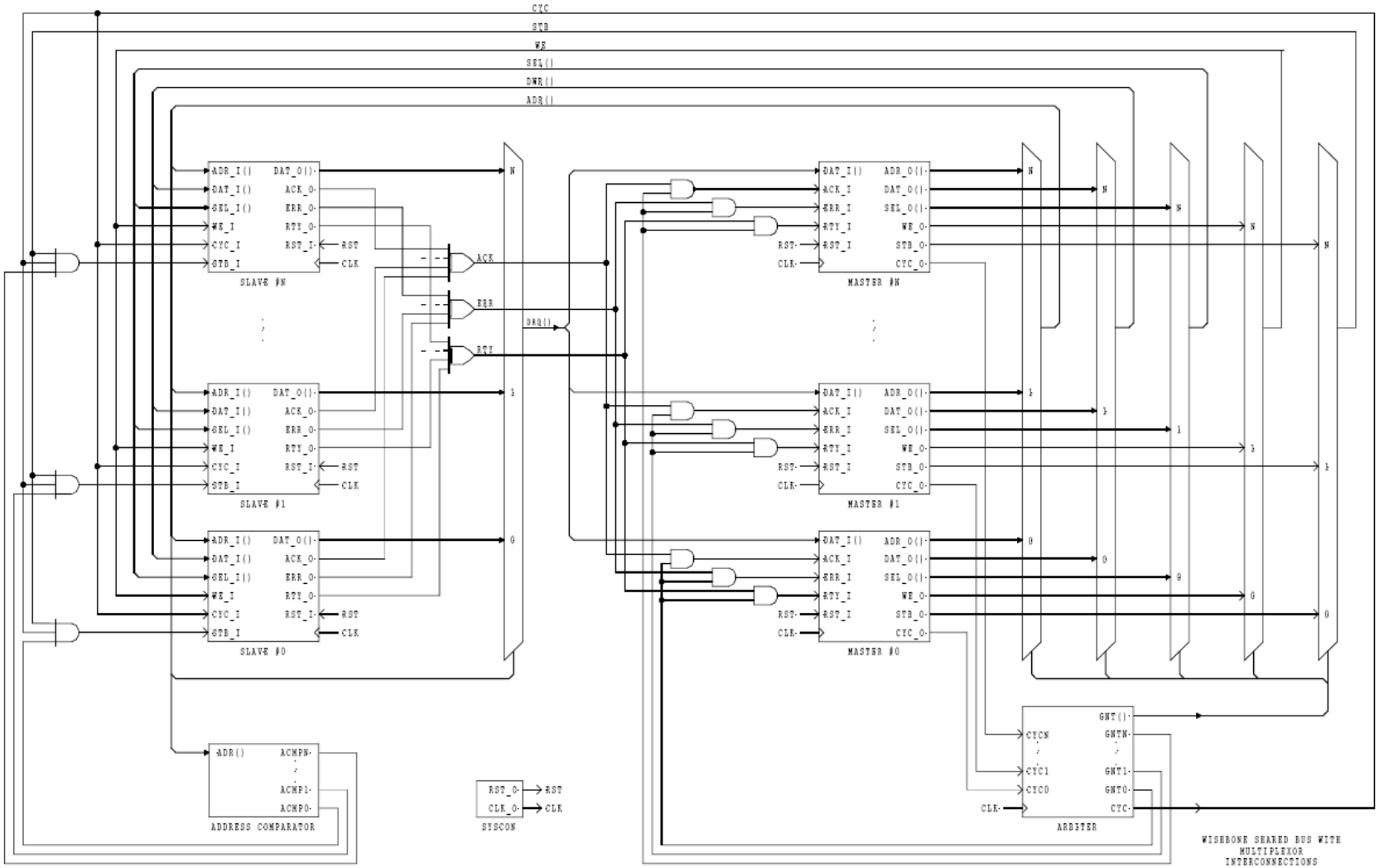
Integration

INTERCON - Adressdekodierung

- Partial Address Decoding:
  - jeder Slave dekodiert nur soviele Adressbits, wie er intern verwendet
  - die oberen Bits werden außerhalb – d.h. im INTERCON – dekodiert



[1], 8.10.4, S. 122f.



[1], Abb. 8-20, S. 121

Integration

Zusammenfassung

- Aufgaben:
  - Cores zusammenstellen und Eigenschaften abgleichen
  - INTERCON definieren
    - Topologie
    - Adresszuordnung
    - Arbitrierungsschema
  - INTERCON und SYSCON bereitstellen
  - Komponenten zusammenführen

Rechtliche Aspekte

Lizenzierung

- „... to provide an open, freely useable interconnect architecture ..."
- „... this document is not copyrighted, and has been placed into the public domain."
- „This specification may be used for the design and production of System-on-Chip (SoC) components without royalties or other financial obligations ..."

Patente

- „The author(s) of this specification are not aware that the information contained herein, nor of products designed to the specification, cause infringement on the patent, copyright, trademark or trade secret rights of others. However, there is a possibility that such infringement may exist without their knowledge. The user of this document assumes all responsibility for determining if products designed to this specification infringe on the intellectual property rights of others."
- Hilfestellung durch Liste bereits betrachteter Patente

[1], S. 3, 89

Zusammenfassung

1. Einführung: Eigenschaften, Architektur
2. WISHBONE-Interface
   - Signale
   - Übertragungsprotokoll
     - Handshaking (Standard Mode)
     - Classic Cycles: Single R/W, Block R/W, Read-Modify-Write
     - Registered Feedback Cycles: Classic, Bursts
3. Integration
   - Freiheitsgrade und WISHBONE Datasheet
   - INTERCON: Topologien, Adressdekodierung, Arbitrierung
4. Rechtliche Aspekte
5. Anhang: Pipelined Mode

Quellen

[0] http://opencores.org/opencores,wishbone

[1] OpenCores: „WISHBONE System-on-Chip (SoC) Interconnection Architecture for Portable IP Cores“, Revision B.4, http://cdn.opencores.org/downloads/wbspec_b4.pdf (Mai 2012)

[2] OpenCores: „WISHBONE System-on-Chip (SoC) Interconnection Architecture for Portable IP Cores“, Revision B.3, 07.09.2002, http://cdn.opencores.org/downloads/wbspec_b3.pdf (Mai 2012)

[3] R. Herveille: „Combining WISHBONE interface signals“, Rev. 0.2, 18.04.2001, http://cdn.opencores.org/downloads/appnote_01.pdf (Mai 2012)

[4] „The WISHBONE Service Center“, http://pldworld.name/_hdl/2/_ip/-silicore.net/wishbone.htm (Mai 2012)

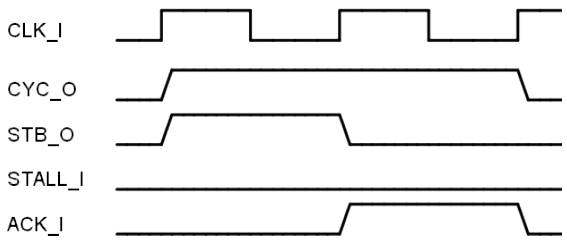
[5] http://en.wikipedia.org/wiki/Wishbone_(computer_bus)

»Wissen schafft Brücken.«

Pipelined Mode

Ansatz

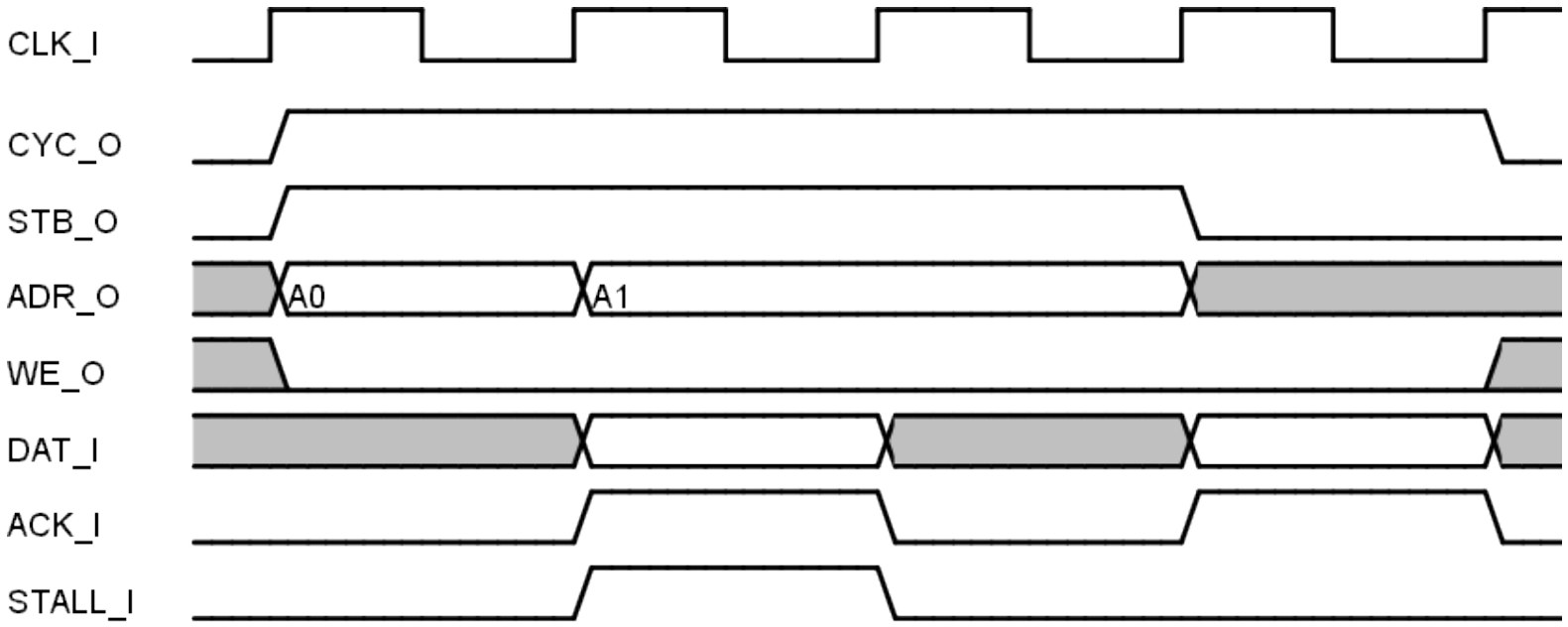
- bei komplexen INTERCONs kann Verzögerung durch Verbindungsnetzwerk kritisch werden (Multiplexerstufen, Routing Delay)
- Entkopplung durch Pipelining wünschenswert
- Problem: solange ein Transfer offen ist, kann kein weiterer ausgelöst werden (STB wartet auf ACK)
  - ➔ max. ein offener Transfer in der Pipeline
  - ➔ jede Pipeline-Stufe verlangsamt Transfer um einen Takt
- Ansatz: Modifikation des Handshaking-Protokolls, so dass Transfer Requests nicht durch ausstehende ACKs blockiert werden



[1], 3.1.3.2, S. 36f.

Pipelined Mode

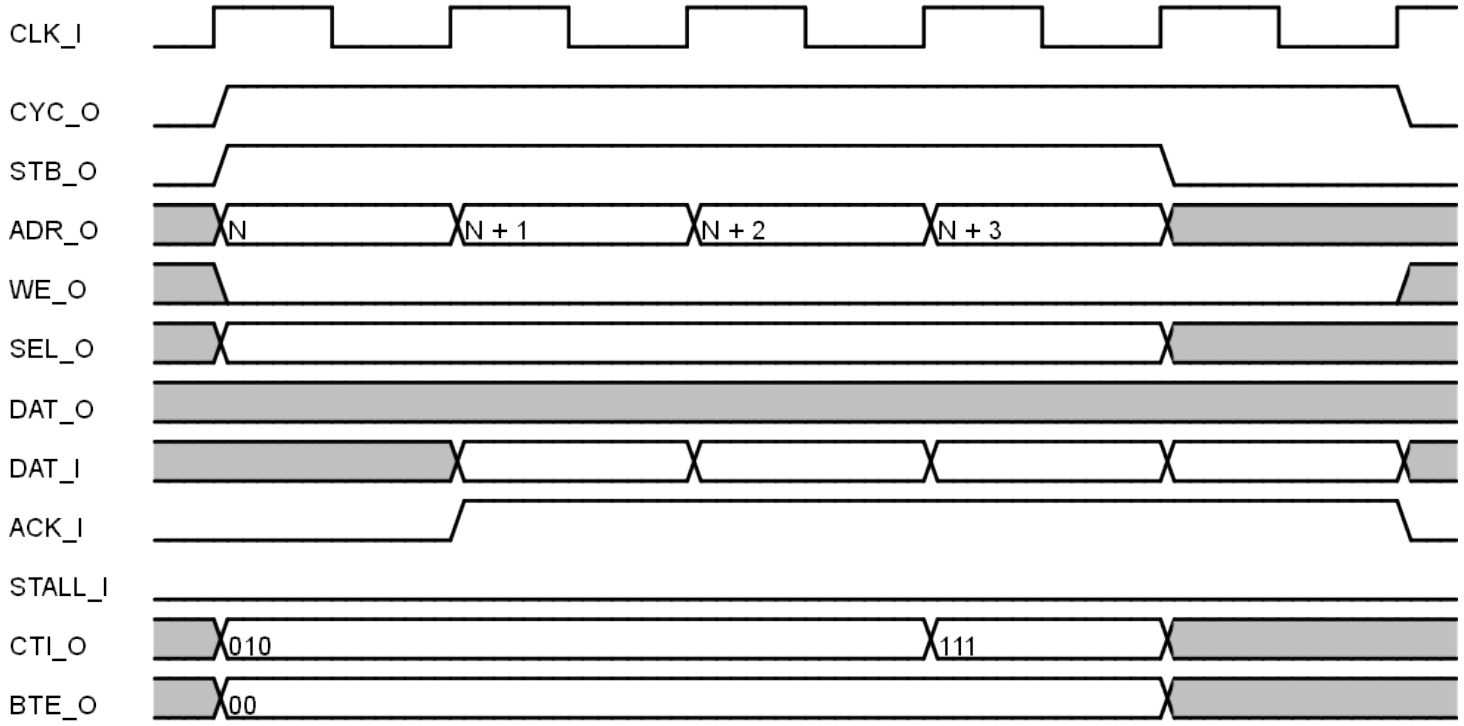
Block Read



[1], 3.3.1, S. 50f.

Pipelined Mode

Incrementing Burst



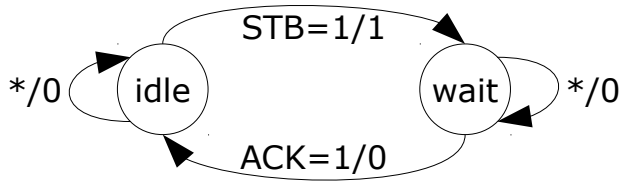
[1], 4.4.4, S. 79ff.

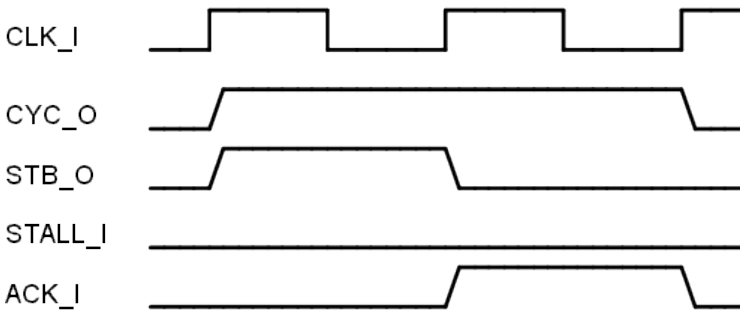
Pipelined Mode

Kopplung von Standard- und Pipelined-Komponenten

Standard-Master an Pipelined-Slave:

STB-Maskierung mittels Zustandsautomat

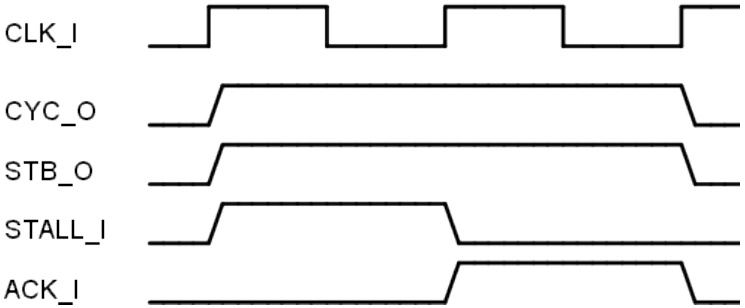




Pipelined-Master an Standard-Slave:

Nutzung von STALL zur Blockierung überlappender Transfers

STALL <= CYC and not ACK;



[1], 5, S. 83ff.

»Wissen schafft Brücken.«