Computertechnik 2 - HS19

Pascal Brunner - brunnpa7

Inhaltsverzeichnis

1 Microcontroller

Ein Microcontroller ist eine SSingle Chip Solution"mit einem CPU und verschiedenen Peripherie-Geräten (bspw. Timer, Counter, Pulse etc.).

1.1 Motivation

Man möchte ein kompaktes System → Embedded Systems

- tiefe Kosten
- Echtzeit
- Einsetzbar in extremene Umgebungen (Hitze, Regen, Schnee etc.)
- zuverlässig
- tiefer Stromverbrauch
- tragbar

1.2 Systembus

Der CPU ist der "Chef"dieses Busses. Der CPU sagt dem Systembus, wann was geschrieben oder gelesen wird. Die Peripheriegeräte und der Speicher agieren nur als SSklaven", welche Anfrage beantworten. Die Bus-Spezifikation sieht wie folgt aus:

- Protokoll und Operationen
- Signale → Anzahl von Signalen, Signalbeschreibung
- Timing → Frequenz, Setup and hold times
- elektrische Eigenschaften
- mechanische Anforderungen

Die Signale-Gruppen werden unterteilt in **Address Linien**:

- geht vom Master zum Sklaven
- Anzahl Linien entspricht der Grösse des Adressspeichers

Daten Linien:

- 8, 16, 32 oder 64 parallele Linien von Daten
- bidirektional (read/write)

Controlsignale:

- Kontrolle read/write Richtung
- provide timing information

1.2.1 Bus Timing Option

Für das Bus Timing gibt es zwei Optionen:
Synchron

- Master und Sklave verwenden die gleiche Clock
- Taktflanken (Clock) steuern Bustransfer auf beiden Seiten
- bei den meisten on-chip Bussen verwendet
- Off-Chip: DDR und synchroner RAM

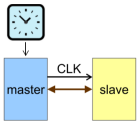

Abbildung 1: Master und Sklave verwenden die gleiche Clock

Asynchron

- Sklave hat keinen Zugriff auf die Taktflanken (Clock) des Masters
- Kontrollsignal speichert die Timing-Information um eine Synchronisation zu gewährleisten
- Vor allem für low data-rate off-chip Speicher gebraucht → paralleler Flash-Speicher und asynchroner RAM
- Off-Chip: DDR und synchroner RAM

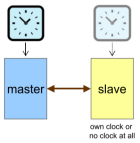

Abbildung 2: Master und Sklave verwenden unterschiedliche Clocks

1.2.2 Mehrere Devices verwenden dieselbe Datenlinie

Mehrere Devices können die gleiche Datenlinien verwenden. Hier kann es zu einem Kurzschluss führen, wenn ein Sklave eine "1ßschreibt und der andere Sklave eine "0ßschreibt. Denn da wird die das Signal beim Master geerdet. Der CPU definiert welcher Sklave den Datenbus in entsprechender Zeit gebrauchen darf.

- Write CPU-driver verbindet mit dem Bus Alle andere Sklaven werden disconnected
- Read CPU-driver nicht verbunden Nur ein einzelner Sklave ist verbunden, alle andere Sklaven sind nicht verbunden.

1.3 Synchroner Bus

Internal workings of the system bus are not disclosed by STM. Die Signalnamen, das Bus-Protokoll und die Timing sind basierend auf der externen Synchronität-Modus. Die **Namenskonvention** bestimmt, dass ein Prefix von "N"(Nxxx) für ein aktives tiefes Signal steht. Des Weiteren steht NOE für *NOT OUTPUT ENABLE*

- NOE = 0 → output enabled
- NOE = 1 → output disabled

Folgende Signale sind für uns relevant:

- CLK
- NE = Not enabled → Gibt einen Hinweis, dass es ein Start / End von einem Kreislauf ist, active-low
- NWE = Not write enabled, active-low
- NOE = Not Output Enable
- NBL = Not Byte Line → Damit erhält man die Möglichkeit Werte in ein WORD zu speichern.

1.4 Control und Status Register

Control Bits

- erlaubt dem CPU die Sklaven zu konfigurieren
- CPU schreibt RegisterBit
- Sklaven-Hardware verwendet den Output der RegisterBit
- Normalerweise read/write

Status Bits

- erlaubt dem CPU die Sklaven zu überwachen
- Sklave schreibt den Status in das RegisterBit
- CPU liest RegisterBit
- normalerweise read

⇒ Das Gleiche Register kann sowohl die Control, wie auch die Status Bits kontrollieren

1.5 Address Decoding

Address Decoding ist die Interpretation der Address-Leitungswerte.
Full Address Decoding

- Alle Addresslinien sind decodiert (checked)
- Ein Control-Register kann genau an einem Ort zugegriffen werden
- 1:1 Mapping → Eine eindeutige Addresse zeigt genau zu einem spezifischen Hardware-Register (physikalischer Speicherort)

Partial Address Decoding

- Nur ein kleiner Teil der Addressen sind decodiert
- erkennt einen Address-Bereich oder ein Set von Addressen
- n:1 Mapping → n eindeutige Addressen zeigen zum gleichen Hardware-Register (physikalischer Speicherort)
- Motivation → Einfacheres decodieren, Verwendung von Alias

1.6 Langsame Sklaven (Slow Slaves)

Durch das Einfügen von "Wait-States"kann man die Verwendung von Slow Slaves verbessern. Eine weitere Möglichkeit ist, dass man eine zusätzliche Kontrollleitung, bei welchem der Sklave sagen kann ïch bin bereit".

1.6.1 volatile

Der Compiler kann in gewissen Situation Variablen oder Statements entfernen, wenn es aus Sicht des Compiler nicht benötigt wird. Mit *volatile* stellt man sicher, dass der Zugriff auf die Variabel jeweils auf den effektiven Wert auf der Hardware zugegriffen wird. Und dass der Compiler diesen Wert nicht reduziert oder verändert. Dadurch wird sichergestellt, dass die Variable sich nicht verändert hat z.B. durch andere Prozesse oder Threads.

2 Memory

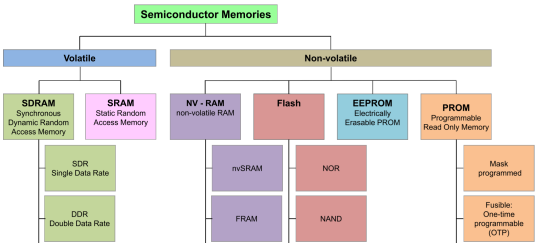

Abbildung 3: Übersicht der einzelnen Speicherarten

2.1 Die unterschiedlichen Einheiten

- **Unit Symbols**
  - $b = bit$
  - $B = Byte$
- **Memory Chips** → Binäre prefixes gemäss JEDEC und IEC
  - Kilo K $= 1024$
  - Mega M $= 1024 * 1024 = 1'048'510$
  - Giga G $= 1024 * 1024 * 1024 = 1'073'741'824$
- **Hard Disks** → Often Use SI (or metric) prefixes
  - Kilo k $= 1000$
  - Mega M $= 1000 * 1000$
  - Giga G $= 1000 * 1000 * 1000$

2.2 CT System im Überblick

ein vereinfachtes Model des STM32F429ZI

- on-chip bus: → 32 Datenlinien, 32 Addresslinien und Kontrollsignale
- external bus: → 16 Datenlinien, 26 Addresslinien und Kontrollsignale

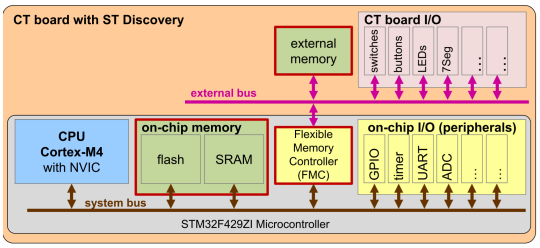

Abbildung 4: CT board mit ST Discovery

2.3 On-chip Speicher: SRAM

SRAM steht für Static Random Access Memory

- Lesen und Schreiben
  - Alle Zugriffe brauchen ungefähr die gleiche Zeit
  - Zugriffszeit ist unabhängig wo die Daten im Speicher liegen
  - Zugriffszeit ist unabhängig vom vorherigen Zugriff
- volatile
  - Speicherinhalt ist nur solange verfügbar wie der Speicher angeschalten ist
- static
  - Speicher Elemente sind identisch zu flip-flops
  - **Kein** refresh notwendig → periodisches Lesen und Umschreiben der Speicherzelle, um den Inhalt zu erhalten.
- Address Regionen
  - SRAM1 112K bytes
  - SRAM2 16K bytes
  - SRAM3 64K bytes
  - CCM 64K bytes

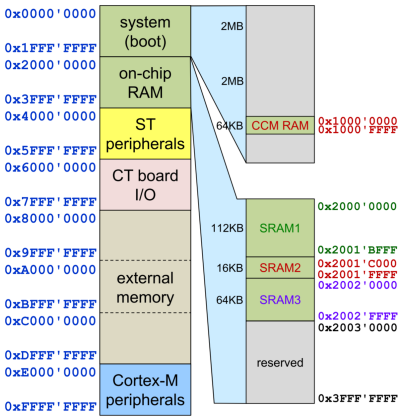

Abbildung 5: Die Address Regionen eines SRAM Speichers

- Schreiben einer Row (Word):
  set bit lines b and !b to (1,0) or (0,1) respectively ⇒ Set the addressed word line to 1 ⇒ Data is stored in cells ⇒ Set word line to 0
- Lesen einer Row (Word):
  pre-charge both bit lines b and !b to 1 ⇒ Briefly set word line to 1 ⇒ inverts pull either b or !b towards (not to ground) ⇒ Sense amplifier amplifies small voltage difference between lines b and !b

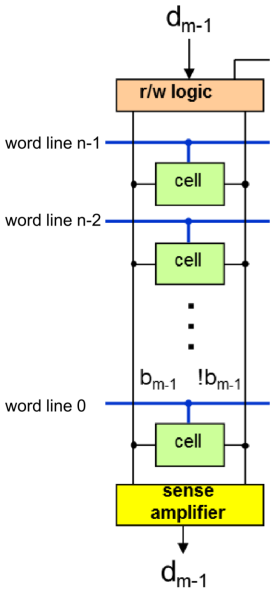

Abbildung 6: Das Schema für das Lesen und Schreiben einer Row (Word) auf einem SRAM

3 GPIO