Institut für Technische Informatik

Entwurf von Echtzeitsystemen

Wintersemester 2014

3. Übungsblatt

Übungsdatum: 14.11.2014

Gruppe: 1

Ausarbeitung von: Allen

Graz, am November 13, 2014

Aufgabe 1

I^2C Anwendungsbeispiel
Sie haben einen Controller gebaut, welcher die Funktion einer Wendeschützschaltung übernimmt. Der Controller verfügt neben für den "Inselbetrieb" notwendigen Eingängen über eine I^2C Schnittstelle. Diese Schnittstelle erlaubt es, zusätzliche Parameter einzustellen, beziehungsweise Betriebszustände (Fehlermeldungen) abzufragen. Erklären Sie

1. Wie sieht ihr Controller aus (Skizze mit Interface)?
2. Arbeitet er als Master oder Slave?
3. Erklären Sie die Register ihres Controllers bzw. deren Funktion?
4. Wie funktioniert die Steuerung des Controllers über I^2C Bus (Beispiel)?

Aufgabe 2

I²C Bus-Arbitrierung
Zeigen Sie anhand vpn Beispielen wie die Arbitrierung am I²C Bus für 2 Master funktioniert, wenn die Master Write-Befehle an Slaves schicken. Geben Sie für folgende Situationen jeweils ein Beispiel an:

1. unterschiedliche Zieladresse und unterschiedliche Daten
2. gleiche Zieladresse und unterschiedliche Daten
3. gleiche Zieladresse und gleiche Daten

Zeigen Sie, welche Daten am Bus gesendet werden und wann sich welcher Master zurückzieht und begründen Sie dies.

Arbitrierung von zwei oder mehreren Mastern auf dem I2C-Bus:

Ein Master darf nur mit einer Übertragung starten wenn der Bus gerade frei ist.
Zwei oder mehrere Master können gleichzeitig starten wenn sie die Start-Condition innerhalb der minimum hold time senden.
Jeder Master kontrolliert bei jedem gesendeten Bit (während die SCL high ist) ob auf der SDA-Leitung der Pegel anliegt den er gesendet hat. Wenn ein anderer Pegel anliegt (nur bei gesendet HIGH aber auf SDA LOW möglich - wired-AND) beendet der Master sofort die Übertragung und der andere Master kann seine Übertragung weiter durchführen.
Die Arbitrierung kann den ganzen Sendevorgang dauern, wenn beide Master an die selbe Adresse die selben Daten senden.

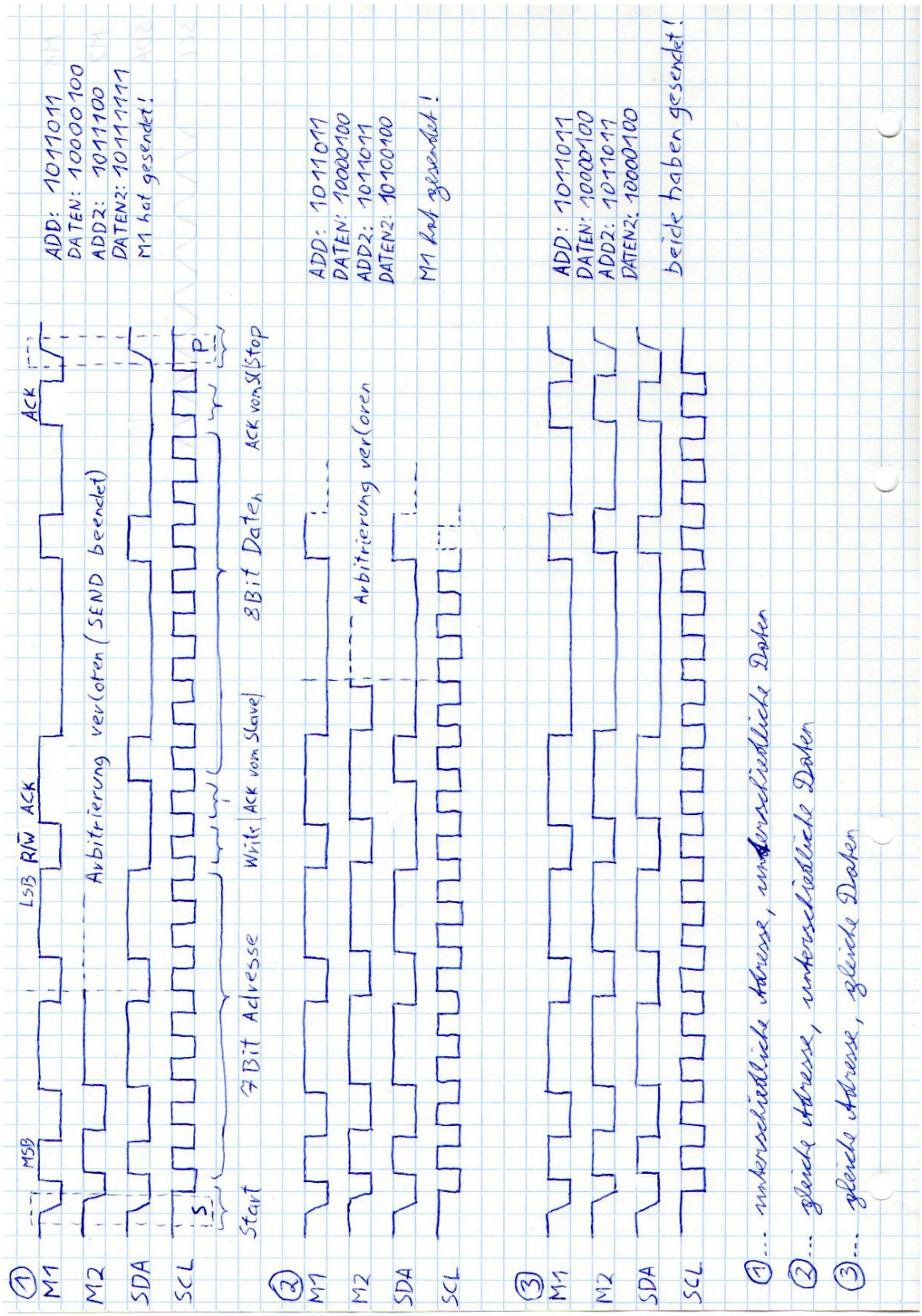

Figure 1: Arbitrierung

Aufgabe 3

I²C-Bus vs. SPI-Bus
Bussysteme spielen eine wichtige Rolle in eingebettenen Systemen. Ermitteln Sie deshalb bedeutende Kenngrößen zu deren Vergleich und identifizieren Sie dies sowohl für den I²C- als auch für den SPI-Bus.
Beschreiben Sie zusätzlich die Zugriffsmethoden für obige Bussysteme und stellen Sie den Ablauf eine Buszyklus dar (Write und Read zwischen Master und Slave). Begründen Sie auch, ob obige Bussysteme mit mehreren Masterknoten verwendet werden können.

Vergleichskriterium	I²C	SPI
Leitungen	2 gemeinsame Leitungen • SDA (Serial Data Line) • SCL (Serial Clock Line)	3 gemeinsame Leitungen • SCLK (serial clock) • MOSI (Master Output, Slave Input) • MISO (Master Input, Slave Output) 1 individuelle Leitung pro Slave • CS (Chip-Select)
Geschwindigkeit	< 400kHz	> 1MHz
Protokollkomplexität	mittel	gering
Chippreise	höher	niedriger
Sendeverfahren	**halbduplex** (Daten können abwechselnd, aber nicht gleichzeitig, in beide Richtungen übertragen werden)	**vollduplex** (Daten können in beide Richtungen gleichzeitig übertragen werden

Table 1: Vergleichskriterien

Der wesentliche Unterschied besteht darin, dass I²C den Busverkehr über 2 Leitungen abhandelt, während SPI die Selektion über eine extra Chip-Select Leitung vornimmt.

I²C verwendet also 2 Leitungen mit 7-Bit Adressierung, jedes Device am Bus kann als Master oder Slave auftreten, es können auch mehrere Master und mehrere Slaves am gleichen Bus betrieben werden.

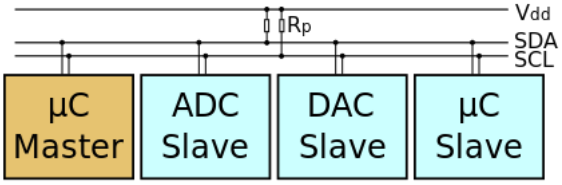

Figure 2: I^{2}C

Ein typischer Sendevorgang sieht folgendermaßen aus: Der Master sendet sein Startbit, gefolgt von der 7-Bit Adresse des Slaves, mit welchem er kommunizieren möchte und einem Bit, ob ein **R**ead oder **W**rite Vorgang stattfinden soll. Ist dieser Slave an dieser Adresse am Bus vorhanden, so sendet er zuerst ein **ACK**(Acknowledge) für diese Adresse, woraufhin der Master in den gewünschten Modus wechselt und der Slave ebenso in den komplementären Modus wechselt (Read—Write oder Write—Read).
Möchte der Master schreiben, so sendet er ein Byte und der Slave sendet ACK, möchte er die Übertragung stoppen, so sendet er entweder ein Stop-Bit, oder erneut ein neues Start-Bit, wenn es weiterhin Kontrolle über den Bus behalten möchte.
Mehrere Master am Bus sind möglich, dabei muss ein Master überprüfen, ob gerade eine Übertragung stattfindet, es werden dabei SDA und SCL beobachtet, ist nur eine LOW, so ist der Bus gerade **busy**. Es kann natürlich auch passieren, das zwei Master *gleichzeitig* senden möchten, hierbei kommt es zu Kollisionsvermeidung, sobald einer der Master High sendet, aber ein Low empfängt, erkennt dieser, dass der Bus soeben benutzt wird und unterbricht seinen Sendevorgang. Der Master, der den Bus erfolgreich arbitriert hat, merkt von den anderen Mastern gar nichts.

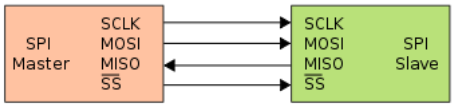

Figure 3: SPI

SPI hat 3 gemeinsame Leitungen und eine individuelle pro Slave, hier darf es nur einen Master geben, der das Clock-Signal an SCLK erzeugt. Mit der CS-Leitung kann der Master einen Slave auswählen, wird die jeweilige CS-Leitungen gegen Masse gezogen, so ist dieser Slave aktiv und wartet an MOSI auf Input und legt selbst seine Daten im Takt von SCLK auf MISO. Es wird dabei ein Byte vom Master an den Slave und ein anderes Byte vom Slave an den Master gesendet.
Mit jeder Taktperiode wird ein Bit übertragen, es sind also normalerweise 8 Taktzyklen notwendig für die vollständige Übertragung eines Bytes notwendig.

Aufgabe 4

ZigBee
Was genau ist ZigBee und worin liegen die Unterschiede zu Bluetooth? ZigBee unterstützt den sogenannten "Beacon Mode". Worum genau handelt es sich dabei und was ist ein Beacon? Wie funktioniert die Übertragung im Beacon Mode? Geben Sie ein Beispiel dazu an?

Aufgabe 5

Schaltnetzwerke
Zeigen Sie wie der Unterschied der Komplexität (Anzahl nötiger Schaltelemente) zwischen einer Corssbar und einer mehrstufigen Butterfly-Schaltnetzwerk zustande kommt. Zur Erinnerung: Ein Crossbar-Switch hat Komplexität $O(n^2)$ im Vergleich zu einem Butterfly-Netzwerk $O(nlog(n)$, um n Knoten miteinander zu verbinden.

Aufgabe 6

CAN Bus
Machen Sie sich mit dem Aufbau eines CAN Daten-Frames vertraut und erarbeiten Sie eine Formel zur Bestimmung der maximalen Netto-Datenrate $R(D, E, p)$ für die Nutzdaten in Abhängigkeit von DLC $D \in \{0, 8\}$ byte, Extenden bit $E \in \{0, 1\}$ und Busgeschwindigkeit in p in bit/s. Geben Sie anschließend eine Formel zur Bestimmung des zugehörigen Protokoll-Overheads $\rho(D, E)$ an.
Bestimmen Sie $R(8, 1, 500000), R(8, 0, 500000)$ sowie alle möglichen Werte für ρ.
Wie muss die Formel für R hinsichtlich der zu erwartenden Netto-Datenrate $R_{exp}(D, E, \rho, \alpha)$ verändert werden, wenn mit einer Paketfehlerwahrscheinlichkeit $\alpha \in [0, 1]$ zu rechnen ist, die jeweils den Neuversand des kompletten betroffenen Daten-Frames erfordert?