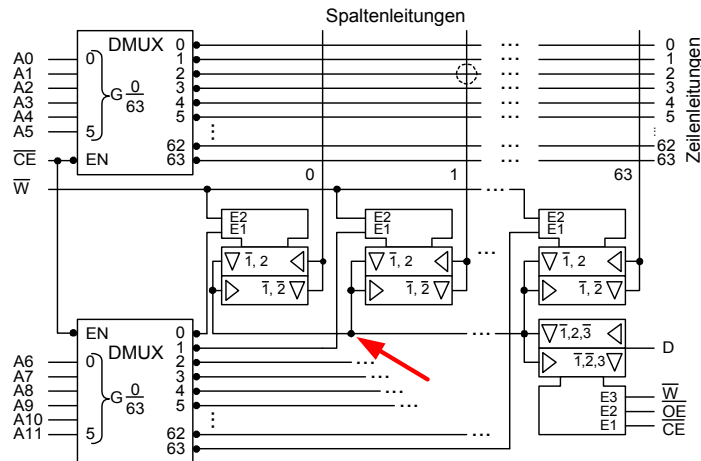


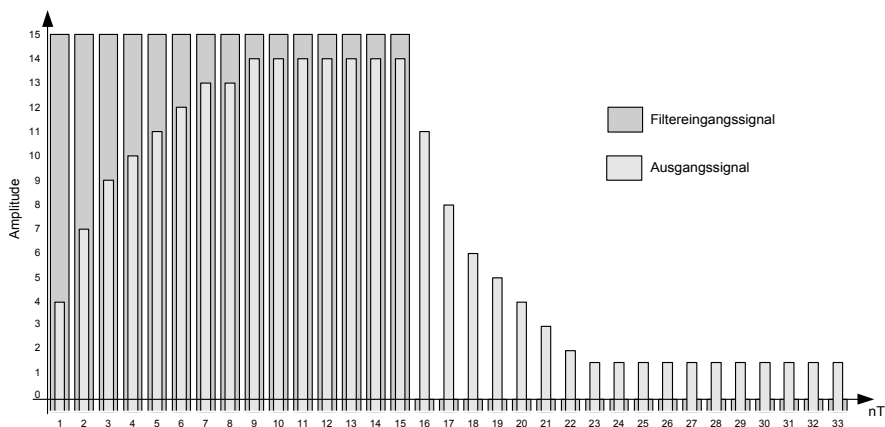
# Bekannte Druckfehler im Buch *Embedded Controller – Eine Einführung in Hard- und Software*

Stand 9. März 2010

- Seite 29 Bild 2.2 Am Ausgang des Multiplexers fehlt ein „D“, um ihn als Datenausgang zu kennzeichnen.
- Seite 35 Tabelle 2.3  $t_{OH}$  output hold time: die angegebenen Zeiten sind *minimal* (nicht max.)
- Seite 41 Bild 2.10 Verbindungspunkt fehlt, s. Bild

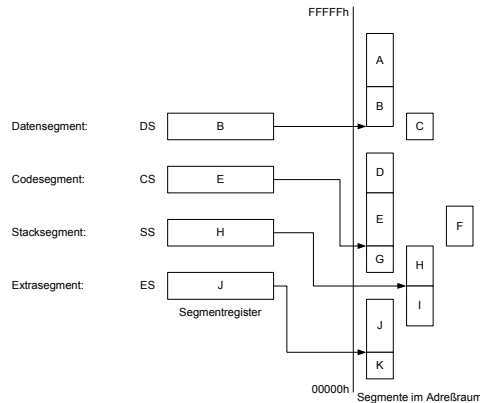


- Seite 55 3. Zeile ersetze *unteren* durch niederwertigeren
- Seite 57 Bild 2.25 Negierungspunkte an den Ausgängen des Registers (Q0 – Q7) sollten entfallen, das Schreiben eines Datums mit einer binären Null und den restlichen Bits auf eins selektiert dann die entsprechende Bank.
- Seite 62 Bild 2.28 Ausgangssignalform paßt nicht zu Tabelle 2.6, (Signal bleibt durch Quantisierung auf 14 bzw. 2)



- Seite 64 Bild 2.30 zweitoberste Datenleitung muß heißen D7 (statt D6)
- Seite 73 Kapitel 3.4, 1. Absatz, 6. Zeile ergänze:  
 ... Die Adressierungsart eines Befehls gibt an, wie und wo die Operanden des Befehls zu holen bzw. zu speichern sind....

- Seite 87 Bild 4.5 ersetze *Basisregister* durch Pointerregister
- Seite 89 Tabelle 4.4, Bit 6 ...ZF wird gesetzt (1), wenn Ergebnis = 0, sonst wird das Zero Flag rückgesetzt
- Seite 92 Bild 4.9 Kasten für Segment E rechts sollte maximale Größe haben, da im Text (S. 91) auf den Offset Overflow Bezug genommen wird

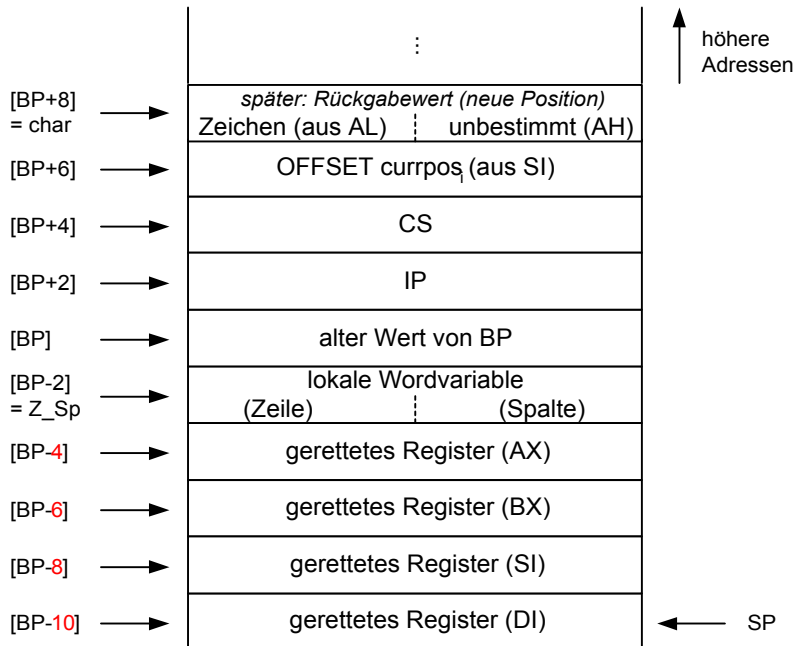


- Seite 93 Tabelle 4.5 BP als Indexregister – alternative Segmentregister: CS, DS, ES
- Seite 128 Bild 4.34 Bezeichnung Empfangsschieberegister: SxRBUF (statt RxBUF)  
Bezeichnung Sendeschieberegister: SxTBUF (statt TxBUF)  
dto.  
dto.
- Seite 129 2. Absatz, 3. Zeile dto.  
3. Absatz, 2. Zeile dto.
- Seite 129 1. Absatz, 2. Zeile dto.  
2. Absatz, 4. Zeile dto.  
3. Absatz, 8. Zeile dto.
- Seite 131 Bild 4.37, „Funktion“ zu Bit TI dto.  
„Funktion“ zu Bit TXE dto.  
„Funktion“ zu Bit OE dto.
- Seite 157 3. Zeile ... eine Variable mit 16 Bit (WordVar1), die mit ...
- Seite 162 Bild 5.5 message\_2 (statt message\_2), u.U. in Codes der CD-ROM ändern
- Seite 162 Bild 5.6 words dw 4731, ... , 10000000**0**b (eine Null zuviel)
- Seite 169 Tabelle 5.5 bei „indirekt“ vor „zus. Displacement möglich“:  
[bp + di] (statt 2x [bp + si])  
bei „indiziert“ vor „zus. Displacement möglich“:  
Label[bp + di] (statt 2x Label[bp + si])
- Seite 182 Tabelle 5.7 *sub* (subtract)  
subtrahiert den Quelloperanden vom Zieloperanden. Das Ergebnis ...  
  
*sbb* ( subtract with borrow)  
subtrahiert den Quelloperanden sowie den Wert des Carry (0 oder 1) vom Zieloperanden. Das Ergebnis ...  
  
*cmp* (compare)  
subtrahiert den Quelloperanden vom Zieloperanden, schreibt das ...

- Seite 188 letzter Absatz, 2. Zeile ... Der TEST-Befehl arbeitet wie das logische UND, verändert aber den Zieloperanden nicht. Mit Ausnahme von NOT verknüpfen sie jeweils zwei Operanden und verändern ...
- Seite 189 Tabelle 5.8
  - and* (logical and)  
verknüpft die beiden Operanden bitweise gemäß einem logischen UND. Das Ergebnis steht im Zieloperanden.  
...
  - test* (test, logical compare)  
verknüpft die beiden Operanden bitweise gemäß einem logischen UND. Das Ergebnis wird *nicht* im Zieloperanden gespeichert. ...
  - or* (logical or)  
verknüpft die beiden Operanden bitweise gemäß einem logischen ODER. Das Ergebnis steht im Zieloperanden, sonst wie *and*.
  - xor* (logical exclusive or)  
verknüpft die beiden Operanden bitweise gemäß einem logischen Exklusiv-ODER. Das Ergebnis steht im Zieloperanden, sonst wie *and*.  
...
  - shl* (shift left)  
schiebt den Inhalt des Register- oder Speicheroperanden um die angegebene Anzahl nach links. Von rechts werden Nullen aufgefüllt. In jedem Teilschritt wandert das jeweilige MSB ins CF.  
...
- Seite 190 Tabelle 5.8
  - shr* (shift right)  
schiebt den Inhalt des Register- oder Speicheroperanden um die angegebene Anzahl nach rechts. Von links werden Nullen aufgefüllt. In jedem Teilschritt wandert das jeweilige LSB ins CF, sonst wie *shl*.  
...
  - sar* (shift arithmetic right)  
schiebt den Inhalt des Register- oder Speicheroperanden um die angegebene Anzahl nach rechts. In jedem Teilschritt wird links das ursprüngliche MSB nachgezogen (vorzeichenrichtiges Schieben) und das jeweilige LSB ins CF geschoben, sonst wie *shl*.
- Seite 196 Tabelle 5.9
  - scas/scasb/scasw* (scan string for byte/word)  
... mit REPZ/REPNE und einer Anzahl in CX kann auf diese Weise das erste Auftreten eines Wertes in einer Zeichenkette gesucht ...  
(Zusatzbemerkung: Mit REPZ/REPE und ... kann das erste Auftreten eines abweichenden Wertes gesucht werden)
- Seite 200 *ret*, 4. Zeile  
dto. vorletzte Zeile streiche Klammer: ... zusätzlich CS vom Stack geladen. Die Angabe ... ersetze: *Aufgabe* durch *Freigabe*
- Seite 208 letzter Absatz, 3. Zeile ... Mit name1 erhält das Unterprogramm den Namen, mit dem es ...
- Seite 213 Bild 5.20 unter „Modul 2“ muß es heißen:           subr1   PROC FAR
- Seite 216 Bild 5.23 unterster Block des Struktogramms:  
Register restaurieren und Rückkehr zum rufenden Programm  
  
dto., erste Textzeile ... (AX, BX, DI) zu Anfang ...  
dto., zweite Textzeile ... für die weitere Verarbeitung nach DI ... (nicht DL)

- Seite 227 Bild 5.30 negative Displacements teilweise falsch [BP- xx]

im Stacksegment:



- Seite 236 Bild 5.39 7. Zeile von unten: xor ah, ah ;AH löschen
- Seite 303 2. Absatz 1. Zeile: MSP~~430~~-... statt (MSP340-...)  
6. Zeile: dto.  
8. Zeile: dto.
- Seite 324 Tabelle C.2: 3. Spalte: 59 3Bh ; (statt 2Bh)
- Seite 328 Tabelle D.1: (Fortsetzung der Befehlsübersicht)  
...  
mov (statt move) move Kap. 5.3.2, S. 175ff  
...

aktualisierte Version der Fehlerliste unter:

<http://www.ostfalia.de/cms/de/pws/bermbach/book>, dort im unteren Bereich Errata.pdf

alternativ:

<http://public.rz.fh-wolfenbuettel.de/~bermbach>, dort Menüpunkt Buch → Errata