Mikrocontrollertechnik MC/ Automatisierungstechnik AU1/ Hard- und Software-Engineering HS1

– Merkblatt zur Klausur –

18. 9. 2014

Themen:
- Grundbegriffe der Rechnerarchitektur
- Grundlagen der Assemblerprogrammierung
- Ausdrucksmittel der Assemblersprache
- Aufbau eines Assemblerprogramms
- Architektur und Befehlsliste der Atmel-AVR-Mikrocontroller
- Die E-A-Ports der Atmel-AVR-Mikrocontroller
- Elementare Programmabläufe

Die Mikrocontrollerprogrammierung in C (vgl. Praktikum) kommt nicht dran.

Wichtige Sachgebiete:
- Initialisierung
- Elementare Ein- und Ausgabe
- Zugriffe auf Daten im Programmspeicher
- Der Stack
- Programmablaufsteuerung
- Elementare Informationswandlungen und Verknüpfungen (wie beispielsweise Verschiebung oder Addition)
- Unterprogrammruf
- elementare Unterbrechungsbehandlung (vgl. Praktikum)

Schwerpunkte im Skript Mikrocontrollertechnik (Seiten gemäß Stand 2014/03):
- 1 bis 28 (Grundlagen, Grundbegriffe),
- 43 bis 46 (Binärzahlen),
- 59 und 60 (Zeichenketten),
- 66 bs 72 (Verschieben, Rotieren, Rechnen),
- 87 bis 96 (Befehlsformate, Registermodelle),
- 115 bis 125 (RISC, CISC, einfache E-A-Ports).

Informationsquellen:
- Skript Mikrocontrollertechnik
- Einführung in die Mikrocontroller-Programmierung am Beispiel Atmel AVR
- Elementare Programmiertechniken, Teil 1
- Die Musterprogramme zum Praktikum
- Die Original-Befehlsbeschreibung der Fa. Atmel (8 Bit AVR Instruction Set). Im Grunde genügt die Kurzübersicht der Seiten 10 bis 15 (Instruction Set Summary). Die komplizierten Verzweigungen kommen nicht dran. Wir brauchen nur BREQ und BRNE.
- Beliebige Lehrbücher zur AVR-Programmierung
- Einschlägiges Material aus dem Internet. (Vorsicht: die Leute in den Foren wissen nicht alles, und was sie zu wissen vorgeben, stimmt zwar manchmal, aber nicht immer. Abhilfe: selber denken...)

Die Klausuraufgaben umfassen:
- Wissensfragen zur Rechnerarchitektur, zu den E-A-Ports und zur Assemblerprogrammierung (beschränkt auf den Stoff, wie er in der Vorlesung und den Übungen behandelt wurde),
- das Schreiben kleiner Programmstücke. Eine besondere Einarbeitung in die Atmel-Peripherie (A/D-Wandler, serielle Schnittstelle usw.) ist nicht erforderlich.

Wichtige Maschinenbefehle

Befehl	Wirkung
LDI *rd, imm*	Lädt einen Direktwert in ein Register. Beispiel: *LDI r17, 0x22*
MOV *rd, rr*	Transportiert Inhalt des Registers rr in das Register rd. Beispiel: *MOV r2, r22*
LPM	Lädt Byte aus Programmspeicher in Register r0. Adresse in Register Z (r31, r30)
OUT *port, rr*	Ausgabe eines Registerinhaltes. Beispiel: *OUT portd, r17*
IN *rd, port*	Eingabe in ein Register. Beispiel: *IN r17, pind*
SBI *port, bit*	Setzen Bit in E-A-Port. Beispiel: *SBI porta, 3*
CBI *port, bit*	Löschen Bit in E-A-Port. Beispiel: *CBI porta, 3*
SBIC *port, bit*	Bit in E-A-Port abfragen. Folgebefehl überspringen, wenn = 0. Beispiel: *SBIC porta, 3*
SBIS *port, bit*	Bit in E-A-Port abfragen. Folgebefehl überspringen, wenn = 1. Beispiel: *SBIS porta, 3*
NOP	keine Wirkung (Leerbefehl). Es wird nur eine Taktperiode verbraucht
JMP *label*	Unbedinge Verzweigung. Beispiel: *JMP anfang*
BRNE *label*	Verzweigen, wenn Ergebnis ungleich Null. Beispiel: *BRNE anfang*
BREQ *label*	Verzweigen, wenn Ergebnis gleich Null. Beispiel: *BREQ ende*
CALL *label*	Unterprogrammruf. Beispiel: *CALL warten*
RET	Rückkehr aus Unterprogramm
PUSH *rr*	Register rr in Stack retten. Beispiel: *PUSH r20*
POP *rd*	Register rd aus Stack laden. Beispiel: *POP r20*
AND *rd, rs*	Register konjunktiv verknüpfen <rd> := <rd> AND <rr>. Beispiel: *AND r12, r3*
ANDI *rd, imm*	konjunktive Verknüpfung mit Direktwert. Beispiel: *ANDI r12*, 0b11110111
OR *rd, rs*	Register disjunktiv verknüpfen <rd> := <rd> OR <rr>. Beispiel: *AND r12, r3*
ORI *rd, imm*	disjunktive Verknüpfung mit Direktwert. Beispiel: *ORI r12*, 0b11110111
COM *rd*	Bitweise Negation (Einerkomplement). Beispiel: *COM r17*
ADD *rd, rs*	Registerinhalte zueinander addieren. <rd> := <rd> + <rr>. Beispiel: *ADD r12, r4*
ADC *rd, rs*	Addieren mit Eingangsübertrag. <rd> := <rd> + <rr>. + CF. Beispiel: *ADC r13, r5*
ADIW *rd, imm*	Addieren Direktwert zu Wort (Register 24, 26, 28, 30). Beispiel: *ADC r24, 12*
SUBI *rd, imm*	Subtrahieren Direktwert. Beispiel: *SBI r16, 0x04*
SUB *rd, rs*	Registerinhalte voneinander subtrahieren. <rd> := <rd> – <rr>. Beispiel: *SUB r12, r4*
SBC *rd, rs*	Subtrahieren mit Eingangsübertrag. <rd> := <rd> – <rr> – CF. Beispiel: *SBC r13, r5*
SBIW *rd, imm*	Subtrahieren Direktwert von Wort (Register 24, 26, 28, 30). Beispiel*: SBIW r26, 10*
CP *rd, rs*	Registerinhalte vergleichen (Subtraktion <rd> – <rs>). Beispiel: *CP r12, r22*
CPI *rd, imm*	Vergleichen mit Direktwert (Subtraktion <rd> – imm). Beispiel: *CPI r16, 0x33*
ROL *rd*	Linksrotieren über CF (CF => Bit 0, Bit 7 => CF): Beispiel: *ROL r17*
ROR	Rechtsrotieren über CF (CF => Bit 7, Bit 0 => CF). Beispiel: *ROR r17*
LSL	Linksverschieben. Bit 7 nach CF, Bit 0 wird mit 0 gefüllt. Beispiel: *LSL r17*
LSR	Rechtsverschieben. Bit 0 nach CF, Bit 7 wird mit 0 gefüllt. Beispiel: *LSR r17*

Achtung: Direktwertverknüpfungen nur mit Registern r16...r31. Weitere Einzelheiten s. Befehlsbeschreibung.