Mikrocomputertechnik

Aufgabe 1 - Prozessor mit Akku-Architektur

Folgender Ausschnitt zeigt die Schleife eines Hochsprachenprogramms.

// Lauflicht

// global variables

byte lights;

// program loop

void loop() {

if (lights == 0) {lights = 128; } // start from 0b10000000

PORTB = lights; // lighting up

lights = lights >> 1; // shift right

delay(100); // wait 100 ms

}

Frage 1.1 (8 Punkte): Übersetzten Sie das Programm in Assemblersprache mit dem Befehls-satz des MCT-Mikrocontrollers. Bitte ergänzen Sie Kommentare zum Ablauf.

Frage 1.2 (4 Punkte): Zum Ablauf auf dem Mikrocontroller muss das Programm in Maschinen-sprache übersetzt werden. Beschreiben Sie den grundsätzlichen Ablauf dieser Übersetzung. Wie werden Befehle und Operanden kodiert? Hinweis: Sie brauchen das Programm hierzu nicht in Maschinensprache zu übersetzen.

Frage 1.3 (8 Punkte): Beschreiben Sie den Ablauf Ihres Programms in einer Tabelle über 10 Takte. Die Tabelle soll folgende Signale zeigen: Clock, PC, PS\_DO, Opcode (Dekoder-Ausgang), Addr, K, ALU-Out, Carry-Bit, Akku, PortB. Die Bezeichnungen entsprechen dem Blockschaltbild des Prozessors in der folgenden Abbildung.

Frage 1.4 (4 Punkte): Folgende Abbildung zeigt den MCT-Mikrocontroller mit Akku-Architektur. Beschreiben Sie den Ablauf der Schiebeoperation lsr über zwei Takte.





Hinweis: Startpunkt ist der dekodierte Befehl am Ausgang des Dekoders. Zeichnen Sie die jeweils aktiven Signale (Leitungen) im Diagramm nach.

Aufgabe 2 - Prozessor mit Register-Architektur

Anstelle des Akkus wird eine Registerbank mit 16 Registern eingeführt. Jedes Register kann als Akku verwendet werden. Folgendes Blockschaltbild beschreibt die Realisierung.



Frage 2.1 (4 Punkte): Beschreiben Sie die wesentlichen Unterschiede zur Akku-Architektur aus Aufgabe 1 (Laden von Registern, ALU-Operationen, Laden und Speichern von Operan-den im Datenspeicher).

Frage 2.2 (4 Punkte): Nennen Sie Unterschiede im Befehlssatz und in der Befehlskodierung (Beispiele: Sprungbefehle, Laden und Speichern von Ergebnissen, ALU-Operationen).

Frage 2.3 (4 Punkte): Gibt es Vorteile, die die Registerarchitektur gegenüber der Akku-Architektur bringt (Geschwindigkeit, Programmierbarkeit)? Lassen sich die Register als Teil des Datenspeichers realisieren?

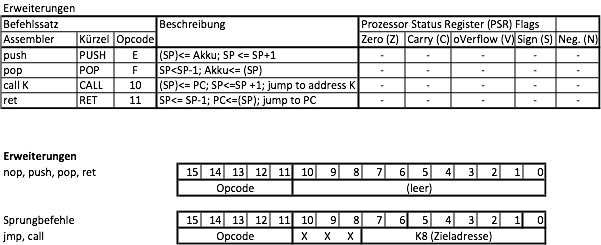
Frage 2.4 (6 Punkte): Schreiben Sie Ihr Programm aus Aufgabe 1.1 auf die Registerarchitektur um. Welche Unterschiede gibt es?

Aufgabe 3 - Erweiterungen des Prozessors (Akku-Architektur)

Zur Unterstützung der strukturierten Programmierung durch Unterprogramme wird der Befehlssatz des Mikrocontrollers durch folgende Befehle erweitert:

* CALL Addr: Ruft Unterprogramm auf.
* RET: Rückkehr aus dem Unterprogramm.
* PUSH: Sichert den Inhalt des Akkus im Datenspeicher.
* POP: Stellt den Inhalt des Akkus aus dem Datenspeicher wieder her.

Die Sicherung des Programmzählers und Akkus im Datenspeicher geschieht in einem gesonderten Bereich (Stapel, engl. Stack), der mit Hilfe eines Stapelzeigers (SP für engl. Stack Pointer) adressiert wird. Folgende Übersicht zeigt die Befehlserweiterung.



Frage 3.1 (8 Punkte): Erläutern Sie die Funktion der Befehle im Detail (Akku, Stapel, PC).

Frage 3.2 (4 Punkte): Welche Befehle werden vom Hauptprogramm aufgerufen? Welche Befehle werden vom Unterprogramm aufgerufen? Skizzieren Sie den Ablauf des Aufrufs eines Unterprogramms in einem Aktivitätsdiagramm.

Text hier eingeben

Frage 3.3 (4 Punkte): Zur Realisierung o.g. Befehle muss der Prozessor erweitert werden. Ergänzen Sie die erforderlichen Erweiterungen im Blockschaltbild.



Frage 3.4 (4 Punkte): Wie viele Takte benötigen die Befehl jeweils: CALL Addr, RET, PUSH, POP. Begründen Sie Ihre Antwort.

Frage 3.5 (4 Punkte): Folgendes Diagramm beschreibt den Stapel im Datenspeicher mit dem SP als Zeiger nach Ablauf eines Befehls. Ergänzen Sie das Diagramm für folgende Schritte: (0) Startpunkt, (1) CALL Addr, (2) PUSH, (3) POP, (4) RET.



Frage 3.6 (10 Punkte): Das Zeitdiagramm in der folgenden Abbildung (Seite 7) beschreibt den Ablauf eines Programms.

* 1. Rekonstruieren Sie den Programmtext aus dem Ablauf (bitte mit Zeilennummern, mit Adressen und Operanden, sowie mit Kommentaren).
  2. Erläutern Sie den Ablauf der Befehle PUSH und POP im Detail (Inhalt Akku und Stapel).
  3. Erläutern Sie den Ablauf der Befehle CALL und RET im Detail (Programmzähler, folgender Befehl, Speicherung und Restaurierung des Programmzählers, Rücksprung an welche Adresse, Ursprung und Zweck des Befehls NOP im Anschluss an CALL).
  4. Erstellen Sie ein Zustandsdiagramm des Prozessors mit den verwendeten Befehlen.

Hinweis: Sie können Zusammenhänge gerne auch im Diagramm mit Pfeilen bzw. Kommentaren markieren.

Seite absichtlich leer - Diagramm zu Aufgabe 3.6 auf der folgende Seite

