



Technische Informatik II im SS 2005

3. Übungsblatt

Abgabetermin: 12. Mai 2005, bis 13:00 Uhr

Prof. Dr. U. Brinkschulte

Geb. 40.28, D-76131 Karlsruhe

Email: brinks@ira.uka.de

Dr.-Ing. T. Asfour

Telefon: +49-721-608-7379

Fax: +49-721-608-8270

Email: asfour@ira.uka.de

<http://i61www.ira.uka.de/users/asfour/TI>

Aufgabe 1

(4 Punkte)

Zur Vereinfachung von Unterprogrammaufrufen wird die MIMA um die Befehle JMS (*jump subroutine*) und JIND (*jump indirect*) erweitert. Der Befehl JMS *ziel* bewirkt, daß die Adresse des nachfolgenden Befehls, also die Rücksprungadresse, an der Speicheradresse *ziel* abgelegt und dann zum Befehl mit der Adresse *ziel + 1* verzweigt wird. Der Befehl JIND *ziel* bewirkt, daß zu derjenigen Adresse verzweigt wird, die in der Speicherzelle *ziel* abgespeichert ist. Dabei werden die vier höchstwertigen Bits des 24 bit langen Speicherwortes *ziel* ignoriert, um eine 20 bit lange Adresse zu erhalten.

OpCode	Mnemonik	Beschreibung
C	JMS a	IAR + 1 → <a>; a + 1 → IAR
D	JIND a	<a> → IAR

Geben Sie die Mikroprogramme für die Ausführungsphase (*execute phase*) der Befehle JMS und JIND an (jeweils ab dem 7. Takt)

Aufgabe 2

(6 Punkte)

Schreiben Sie ein MIMA-Programm, das den Wochentag zu einem beliebigen (TAG, MONAT) im Jahr 2004 berechnet. Dazu werden die Wochentage wie folgt definiert:

0 = Sonntag, 1 = Montag, 2 = Dienstag, 3 = Mittwoch,
4 = Donnerstag, 5 = Freitag, 6 = Samstag

Das Programm soll an der Speicheradresse 0x00100 beginnen. Der TAG steht in der Speicherzelle 0x00020 zur Verfügung und der MONAT steht an Adresse 0x00021 bereit. Nach Ablauf des Programms soll sich die Wochentagszahl an Adresse 0x00030 befinden.

Sie können sich bei der Berechnung an folgendem c-Code orientieren:

```

main() {
    int ersterTag[]={-1,4,0,1,4,6,2,4,0,3,5,1,3};
    int TAG = 3;
    int MONAT = 6;
    int wochentag = TAG + ersterTag[MONAT] - 1;

    wochentag %= 7;

    printf("Wochentag = %i\n", wochentag);
}

```

Die jeweiligen Wochentage des ersten Tages im Monat stehen hier im Array `ersterTag` zur Verfügung. Addiert man zu diesem Wochentag den $(TAG-1)$ und rechnet modulo 7, so erhält man den gewünschten Wochentag wie oben definiert. Beachten Sie hierbei, dass in c die Arrayelemente mit 0 beginnend nummeriert werden, während der Kalender immer bei 1 anfängt zu zählen.

Aufgabe 3

(10 Punkte)

Ziel der Aufgabe ist der Entwurf des Steuerwerkes eines Schnittstellenbausteins für die Anbindung des MiMa-Prozessors an den Systembus. Der Schnittstellenbaustein soll eine geeignete Verbindung zwischen dem Ein-/Ausgabeverhalten der MiMa-Architektur und den Anforderungen, die der Systembus durch seine Spezifikation stellt, ermöglichen.

Der MiMa-Prozessor besitzt folgende Anschlüsse¹:

- R: Read (o) - Daten sollen gelesen werden. Die Adresse liegt bei $A_{19} - A_0$ an.
- W: Write (o) - Es sollen Daten geschrieben werden.
- A: Acknowledge (i) - Schreib- oder Lesevorgang wurde vollständig ausgeführt.
- $A_{19} - A_0$: Adreßleitungen (o)
- $D_{23} - D_0$: Datenleitungen (i/o)

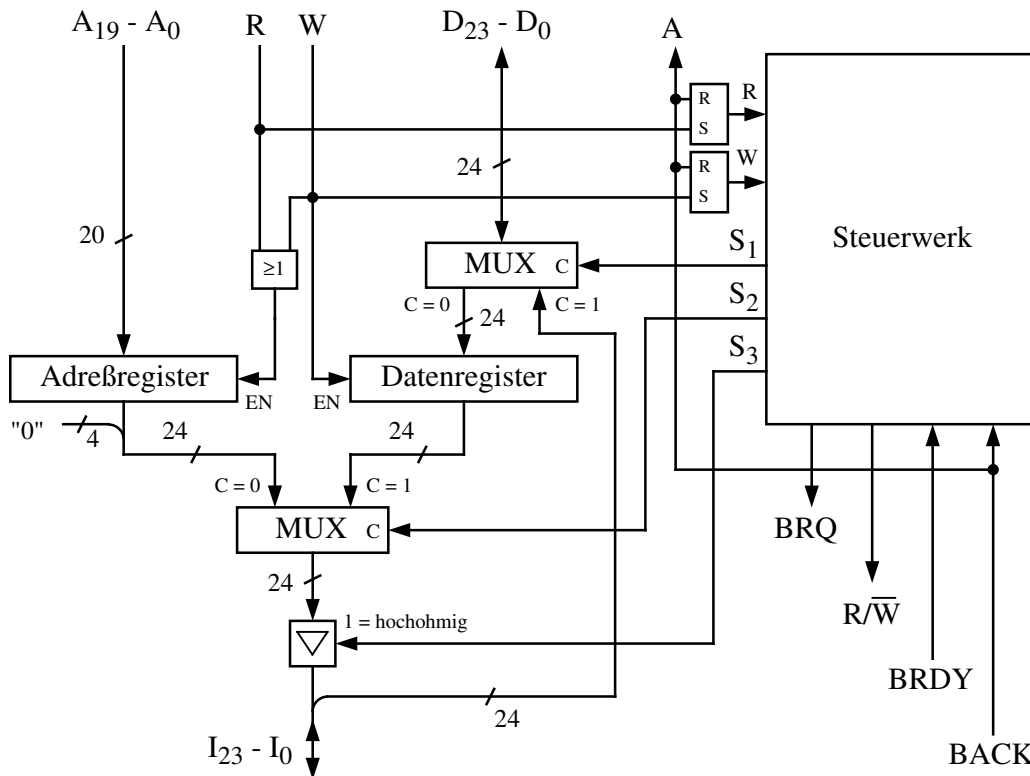
Bei einem Schreibvorgang wird für genau einen Takt $W = 1$ gesetzt, während die gültige Adresse bei $A_{19} - A_0$ und die gültigen Daten bei $D_{23} - D_0$ anliegen. Die Durchführung des Schreibvorganges wird dem Prozessor durch $A = 1$ für genau einen Takt vom Schnittstellenbaustein bestätigt. Bei einem Lesevorgang wird für genau einen Takt $R = 1$ gesetzt, während die gültige Adresse bei $A_{19} - A_0$ anliegt. Die gelesenen Daten liegen an, wenn vom Schnittstellenbaustein für genau einen Takt $A = 1$ gesetzt wird.

Beim hier verwendeten Systembus werden die Adresse und die Daten auf denselben Leitungen im Multiplexverfahren angelegt. Der Systembus besitzt folgende Anschlüsse:

- BRQ: Bus Request (i) - Anforderung an den Bus
- R/\overline{W} : Read/Write (i) - Bus lesen oder schreiben
- BRDY: Bus Ready (o) - Bus steht nach Anforderung zur Verfügung
- BACK: Bus Acknowledge (o) - Bus bestätigt den Empfang von Adresse oder Daten
- $I_{23} - I_0$: Adress- und Datenleitungen (i/o)

¹i = Eingabe, o = Ausgabe, i/o = Ein-/Ausgabe

Bei einer Schreib- oder Leseanforderung an den Bus wird zunächst $BRQ = 1$ gesetzt. BRQ bleibt auf 1 bis der Bus seine Bereitschaft mit $BRDY = 1$ meldet. Im darauffolgenden Takt muß $R/\overline{W} = 1$ für Lesen bzw. $R/\overline{W} = 0$ für Schreiben gesetzt und die Adresse auf den Bus gelegt werden. Der Bus bestätigt den Empfang der Adresse mit $BACK = 1$; R/\overline{W} soll bis dahin gültig bleiben. Bei einer Leseanforderung werden im darauffolgenden Takt die gelesenen Daten vom Bus für zwei Takte bereitgestellt; im Fall einer Schreibanforderung werden im darauffolgenden Takt die zu schreibenden Daten vom Schnittstellenbaustein für zwei Takte auf den Bus gelegt. Der Systembus besitzt für jede angeschlossene Einheit, die Anforderungen (Bus Requests) stellen darf, eine eigene BRQ - und $BRDY$ -Leitung.



Obiges Bild zeigt das Operationswerk des Schnittstellenbausteins. Bei einem Schreib- oder Lesevorgang wird die Adresse durch $EN = 1$ in das Adreßregister geschrieben. Bei einem Schreibvorgang werden die Daten durch $EN = 1$ in das Datenregister übernommen; dazu muß $S_1 = 0$ sein. Durch zwei Flipflops wird festgehalten, ob es sich um einen Schreib- oder Lesezugriff handelt. Die Flipflops werden durch $BACK = 1$ zurückgesetzt und sind im Grundzustand nicht gesetzt. Der obere Multiplexer ist mit Hilfe von Pass-Transistoren so realisiert, daß die über den Eingang C geschalteten Verbindungen bidirektional sind; der jeweilig nicht-geschaltete Ein-/Ausgang ist hochohmig. Je nach Ansteuerung des unteren Multiplexers mit S_2 werden die Daten oder es wird die Adresse auf den Bus gelegt. Mit $S_3 = 0$ wird der Tri-State-Bustreiber durchgeschaltet. Wenn Daten vom Bus gelesen werden sollen, muß der Tri-State-Bustreiber mit $S_3 = 1$ in einen hochohmigen Zustand versetzt werden.

1. Zeichnen Sie ein Zeit-Diagramm für einen Schreib- und einen Lesezyklus des Systembusses.
2. Entwerfen Sie das Zustandsübergangsdiagramm für das Steuerwerk (Moore-Automat).