

Aufgabe 42: (K) Entwurf eines Schaltnetzes

(1+1+2 Pkt.)

In einer Gefahrenmeldeanlage sollen drei Gefahrentypen durch drei Lämpchen angezeigt werden. Spricht nur einer der drei Melder (a, b, c) an, soll die gelbe Lampe G leuchten ($G = 1$). Melden zwei Melder gleichzeitig, soll die orange Lampe O leuchten ($O = 1$) und nur wenn alle drei Melder Alarm geben, soll die rote Lampe R aufleuchten ($R = 1$).

- Stellen Sie die Funktionstabelle der Gefahrenmeldeanlage auf.
- Geben Sie eine Funktionsgleichung der gelben Lampe an.
- Zeichnen Sie ein Schaltbild für den Ausgang G.

Aufgabe 43: (H) 2er-Komplement-Darstellung

(10 Pkt.)

Schreiben Sie ein MIPS-Assembler-Programm, das eine positive oder negative Dezimalzahl einliest und ihre Binärdarstellung unter Verwendung der 2er-Komplement-Darstellung ausgibt. Testen Sie Ihr Programm mit den Eingaben 25 und -25.

Aufgabe 44: (T) Sprungbefehle in Pipeline

(2+1+1+2 Pkt.)

In dieser Aufgabe soll die Auswirkung von Sprungbefehlen auf die in der Vorlesung angegebene fünfstufige Pipeline (*IF, ID, EX, MA, WB*) untersucht werden.

- Skizzieren Sie den Zustand der Pipeline bei der Ausführung von Instruktionen, die auf einen Sprungbefehl folgen. Betrachten Sie einen bedingten Sprung und nehmen Sie dabei an, daß noch keine Maßnahmen zur Vermeidung von Hazards getroffen wurden.
- Leiten Sie daraus Möglichkeiten ab, *pipeline stalls* für diese Situation zu vermeiden bzw. die Anzahl der *stall cycles* zu verringern. Pipeline stalls bzw. pipeline cycles sind Wartezyklen, die z.B durch Einfügen von NOP-Anweisungen entstehen.
- Bei einem *delayed branch* wird die auf einen Sprungbefehl folgende und vom Sprungbefehl unabhängige Anweisung abgearbeitet, bis das Sprungziel bekannt ist. Wie könnte der folgende Code modifiziert werden, um Nutzen aus einem *delayed branch* zu ziehen?

```

Loop:   lw      $2, 100($3)
2       addi    $3, $3, 4
        beq     $3, $4, loop

```

- Gegeben sei folgende Instruktionssequenz:

```

                sub      $10, $4, $8
2             beq      $1, $3, jump    # PC-relative Verzweigung zu Adresse 72
                and      $12, $2, $5
4             or       $13, $2, $6
                add      $14, $4, $2
6             sub      $15, $6, 7
                ...
8 jump:      lw       $4, 50($7)

```

Beschreiben Sie das Verhalten der Pipeline, wenn der Sprungbefehl durchgeführt bzw. nicht durchgeführt wird.