



10. Übungsblatt Informatik II

(Abgabe: 11.07.2003)

1. Aufgabe: (RAM mit clear-Signal) (5 Punkte)

Konstruiere ein $(2^a \times n)$ -RAM mit einem zusätzlichen Eingang *clear*, so dass gilt: Wenn im Takt t das Signal *clear* aktiv ist, dann gilt im Takt $t + 1$: $D_{out}(x) = 0^n \forall x \in \{0, 1\}^a$.

2. Aufgabe: (Primzahlsieb des Eratosthenes) (20 Punkte)

Möchte man alle Primzahlen bis zur Zahl x finden, so kann man wie folgt vorgehen:

Man streiche in einer zusammenhängenden Liste von natürlichen Zahlen, die bei der 2 beginnt und bei x endet, alle "echten" Vielfachen von 2 (d.h. 4, 6, 8,...). Sodann streiche man alle echten Vielfachen der 3 (6, 9, 12,...). Man fahre mit der nächsten noch stehenden Zahl (das ist die 5, da die 4 als Vielfaches von 2 bereits gestrichen wurde) analog fort und wiederhole den Vorgang, bis in der Liste keine Zahl mehr ein Vielfaches einer anderen ist.

Implementiere den Algorithmus in RTL-Sprache wie folgt:

- Zuerst fülle alle Speicherzellen von 2 bis 1000 mit dem Wert 1.
- Am Ende des Algorithmus soll in genau den Speicherzellen zwischen 2 und 1000 eine 1 stehen, deren Adressen Primzahlen sind.

3. Aufgabe: (Aufzugsteuerung) (15 Punkte)

Zu entwerfen ist ein Automat, der die Motoren eines Aufzugs steuert. Dieser soll folgendes Verhalten zeigen:

- Der Automat hat die Eingangssignale $r[1 : 0]$ und die Ausgangssignale *up*, *down* und *open*, wobei immer nur maximal eines der Ausgangssignale gleichzeitig aktiv sein darf.
- Ist das Signal *up* aktiv, so bewegt sich der Aufzug zum nächsten Takt hin um ein Stockwerk nach oben, bei *down* entsprechend nach unten. Bei aktivem *open*-Signal werden bis zum nächsten Takt die Türen geöffnet und wieder geschlossen.
- In jedem Takt können beliebig viele der r_i -Signale aktiviert werden. Diese signalisieren, dass der Aufzug sich in nächster Zeit in das Stockwerk i begeben und dort die Tür öffnen und schließen soll.
- Die r_i -Signale werden akkumuliert und erst dann wieder gelöscht, wenn der Aufzug auf Stockwerk i seine Tür geöffnet und geschlossen hat.

Tipp: Benutzt man einen Automaten mit $2^3 = 8$ Zuständen, so kann man im Zustandsnamen codieren, in welchem Stockwerk man sich gerade befindet und welche der r_i -Signale sich in der Zwischenzeit akkumuliert haben.

4. * Aufgabe: (Write Back Caching) (10 Punkte *)

Beim aus der Vorlesung bekannten Write Through Caching wird jede Änderung im Cache direkt in das MM geschrieben, sodass der Cache eine Kopie des MM bleibt. Benutzt man jedoch nur einen einzelnen Cache, so kann man die Performance des Caches wie folgt erhöhen:

Zu jeder Cacheline wird neben dem Valid-Bit noch ein zusätzliches Dirty-Bit eingeführt. Dieses Bit zeigt an, dass die Cacheline aktueller ist als entsprechende MM-Eintrag. Wird nun eine Cacheline vom Prozessor geändert, so wird sie nicht direkt ins MM zurückgeschrieben, sondern nur das Dirty-Bit gesetzt. Erst dann, wenn eine Line aus dem Cache rausgeworfen wird, weil eine neue Line per Linefill an die gleiche Stelle im Cache geladen werden soll, wird der Cacheinhalt bei gesetztem Dirty-Bit in das MM zurückgeschrieben.

Spezifiziere den neuen Kontrollautomaten und die nötigen Änderungen an der Hardware (wie z. Bsp. die Umgebung des Dirty-Speichers).