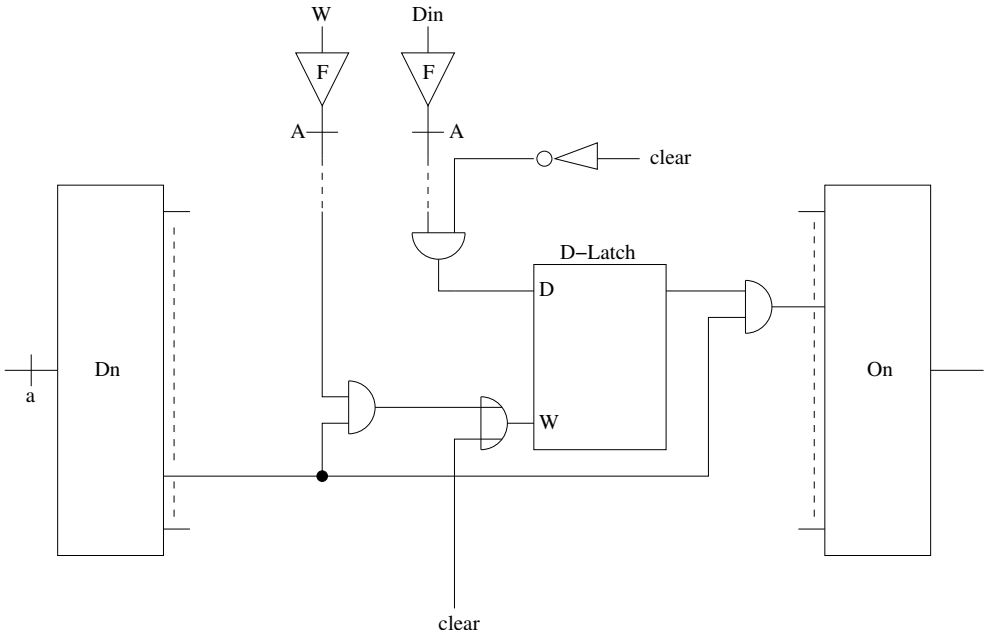
Universität des Saarlandes
FR 6.2 - Informatik
Prof. Dr. W.J. Paul
Dipl. Inform. M. Klein
Dipl. Inform. T. In der Rieden

10. Übungsblatt Informatik II

(Abgabe: 11.07.2003)

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

Das folgende Bild zeigt ein (A x 1)-SRAM (mit $A = 2^a$), das der im Buch von Keller/Paul präsentierten Konstruktion folgt. Um daraus ein (2^a x n)-RAM zu erhalten, muss man lediglich n dieser Bauteile zusammenschalten (mit gemeinsamen Adresseingängen).



Die Modifikationen zum Einbau des *clear*-Signals sind folgende:

- Die Schreibsignale W der einzelnen D-Latches werden mit dem *clear*-Eingang verodert, damit im Falle des Aktivierens von *clear* in alle Speicherzellen gleichzeitig geschrieben werden kann. Falls *clear* inaktiv ist, so wird nur im Falle eines gesetzten W an genau die richtige Speicherstelle geschrieben (wie bisher).
- Die Dateneingänge D der D-Latches werden mit einem negierten *clear*-Signal verundet, damit im Falle des Aktivierens von *clear* in alle D-Latches der Wert 0 geschrieben werden kann. Bei deaktiviertem *clear* verhält sich das SRAM wie gewohnt.

Wenn das *clear*-Signal im Takt t gegeben wird, so wird im selben Takt in alle Speicherzellen der Wert 0 geschrieben. Einen Takt später steht der neue Inhalt dann an allen Ausgängen des RAMs bereit.

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

a)

Wir füllen zuerst alle Speicherzellen von 2 bis 1000 mit dem Wert 1.

Speicherzelle	1	2	$\cdots$	1000	$\cdots$
Speicherinhalt	·	·	$\cdots$	·	$\cdots$

Universität des Saarlandes
FR 6.2 - Informatik
Prof. Dr. W.J. Paul
Dipl. Inform. M. Klein
Dipl. Inform. T. In der Rieden

10. Übungsblatt Informatik II

(Abgabe: 11.07.2003)

0	:	R1 = R0 + 2	Lade die Adresse der zuerst bearb. Speicherzelle zu R1
4	:	M(R1 + 0) = 1	Speichere den Wert 1 in die aktuelle Adresse
8	:	R2 = (R1 = 1000 ? 1 : 0)	Markiere R2 mit 1 wenn wir fertig sind
12	:	R1 = R1 + 1	Inkrementiere R1
16	:	PC = PC + (R2 = 0 ? -12 : 4)	Alle Adressen bearbeitet ⇒ terminiere, sonst gehe zurück
20	:	STOP	

Speicherzelle	1	2	$\cdots$	1000	$\cdots$
Speicherinhalt	·	1	$1 \cdots 1$	1	$\cdots$

b)

Jetzt verwenden wir den *Primzahlsieb des Eratosthenes*. Ein pseudocode dafür kann man so schreiben:

i=2; **while** i $\leq$ 1000 **do**
j=i; (**while** j $\leq$ 1000 **do** j = j+i; **if** j$\leq$1000 **then** M(j)=0);i++;

Wobei man mit der Laufvariable i alle Speicherplätze durchläuft (erste *while*-Schleife) und dann mit j alle echte Vielfachen von i mit 0 ersetzt(zweite *while*-Schleife). Mit einer *if*-Anweisung vermeidet man eine wahrscheinliche Manipulation einer Speicherzelle, deren Adresse grösser als 1000 ist. Bei dem RTL-Programm steht der Register R1 für i und R2 für j.

24	:	R1=R0+2	i = 2
28	:	PC = PC + (R1 $\leq$ 1000 ? 4 : -8)	Wenn i$\leq$ 1000 mach weiter, sonst terminiere.
32	:	R2 = R1 + 0	j = i
36	:	PC = PC + (R2 $\leq$ 1000 ? 4 : 20)	Wenn j$\leq$1000 mach weiter, sonst
40	:	R2 = R2 + R1	j = j + i
44	:	PC = PC +(R2 $\leq$ 1000 ? 4 : 8)	*if*-Anweisung
48	:	M(R2) = 0	M(j) = 0
52	:	PC=PC - 16	gehe zum Anfang der inneren Schleife
56	:	R1 = R1 + 1	i++
60	:	PC = PC - 32	gehe zum Anfang der ausseren Schleife

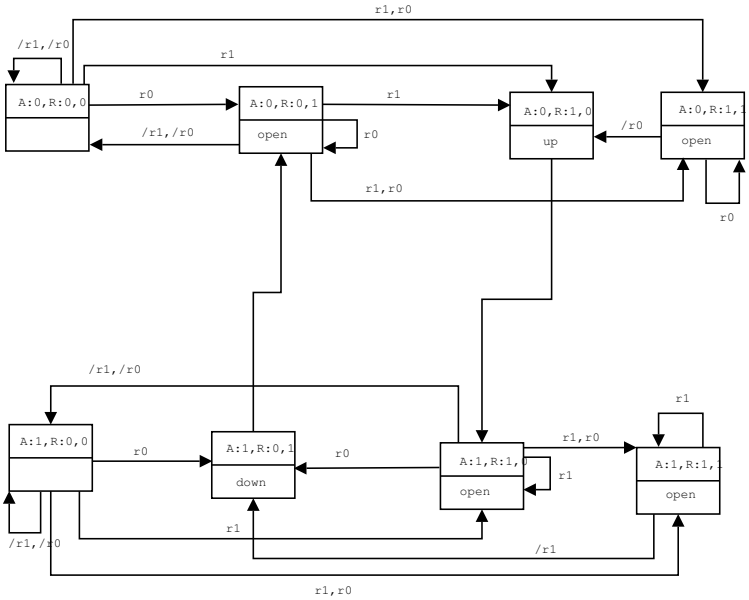
3. **Aufgabe: (Aufzugsteuerung)** **(15 Punkte)**

Die Aufzugsteuerung kann man wie unten gezeigt realisieren. Der Name jedes Zustandes soll sich lesen als: A: aktuelle Etage R: welche Ruftasten wurden betätigt, hierbei ist die linke Zahl r_1, die rechte r_0 Kanten an denen keine Markierungen sind, werden grundsätzlich ausgeführt.

Universität des Saarlandes
FR 6.2 - Informatik
Prof. Dr. W.J. Paul
Dipl. Inform. M. Klein
Dipl. Inform. T. In der Rieden

10. Übungsblatt Informatik II

(Abgabe: 11.07.2003)

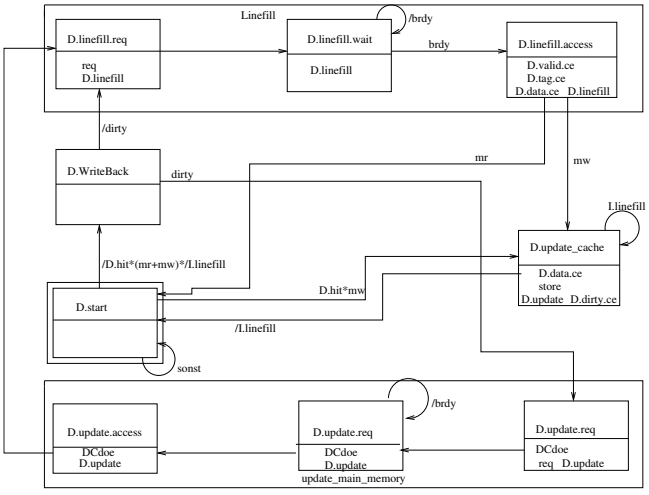


4. * **Aufgabe: (Write Back Caching)** **(10 Punkte *)**

Man fügt einen neuen Zustand (D.WriteBack) im alten Daten-Cache Automaten hinzu. In diesem Zustand wird überprüft,ob das Dirty-Bit gesetzt ist. Wenn:

-ja: dann geht man zu memory-update und erst danach zu linefill;

-nein: dann geht man wie früher in Linefill. Falls das mw=1 ist geht man danach zu cache-update (dort wird das Dirty-Bit gesetzt) und dann zur Start-Zustand zurück.



Zur Hardware wird noch ein RAM (Dirty-RAM) eingefügt.

Die neue Hardware sieht dann so aus:

Universität des Saarlandes
FR 6.2 - Informatik
Prof. Dr. W.J. Paul
Dipl. Inform. M. Klein
Dipl. Inform. T. In der Rieden

10. Übungsblatt Informatik II

(Abgabe: 11.07.2003)