



**Aufgabe 1: (Sättigungszähler)**

( 4 Punkte)

In der Vorlesung haben wir Sättigungszähler (saturating counters) eingeführt. Ein  $n$ -Bit Sättigungszähler ist ein Schaltkreis mit Eingängen *reset*, *inc*, *dec* und Ausgang  $o[n-1 : 0]$ . Der Ausgang  $o$  ist definiert durch:

$$o = \begin{cases} 0^n : \text{reset} = 1 \\ o + 1 : \text{inc} = 1 \wedge o \neq 1^n \\ o - 1 : \text{dec} = 1 \wedge \text{inc} = 0 \wedge o \neq 0^n \\ o : \text{sonst} \end{cases}$$

Entwerfen Sie einen  $n$ -Bit Sättigungszähler. Dazu können Sie einen  $n$ -Bit Standardzähler mit *reset*, *inc* und *dec* Eingängen benutzen.

**Aufgabe 2: (Return Stack Adress Berechnung)**

( 4 + 2 Punkte)

In der Vorlesung haben wir einen sogenannten Return Stack benutzt, um die Zieladressen von returns zu berechnen. Zum Zugriff auf den Return Stack gibt es die beiden Signale *push* (fügt einen neuen Eintrag im Stack hinzu) und *pop* (entfernt einen Eintrag vom Stack). Wenn mehr Adressen auf dem Stack abgelegt werden, als Platz vorhanden ist, werden die ältesten Einträge überschrieben.

- In der Vorlesung haben wir den Return Stack mit Hilfe eines Standard Zwei-Port-RAMs implementiert. Wie müssen die Schreib- und Lese-Adressen für das RAM berechnet werden, wenn ein Register *rs.addr* zum Speichern der Adresse des aktuell obersten Elementes des Stack vorhanden ist?
- Entwerfen Sie einen Schaltkreis der ein Register für die Return Stack Adresse *rs.addr* enthält und abhängig von den Signalen *pop* und *push* den neuen Inhalt von *rs.addr* berechnet. Der Schaltkreis soll ebenfalls die Lese- und Schreibadresse (*rs.r.a* bzw. *rs.w.a*) ausgeben.

**Aufgabe 3: (Optimierter Forwarding Schaltkreis)**

( 3 + 5 Punkte)

In Abbildung 1 sehen Sie den Original Forwarding Schaltkreis aus der Vorlesung. Der Or(5)-Baum wird benutzt um Zugriffe auf Register GPR[0] zu erkennen und zu behandeln. Statt dessen kann man die GPR[0]-Erkennung auch wie in Abbildung 2 lösen.

- Berechnen Sie die Kosten und das Delay des neuen Forwarding Schaltkreises und vergleichen Sie es mit den Originalwerten.
- Mit dem neuen Forwarding Schaltkreis ist es möglich die Implementierung des GPR Environments abzuändern. Insbesondere bei der Behandlung von GPR[0] kann Hardware eingespart werden. Welche Änderungen sind möglich? Vergleichen Sie Kosten und Delay des modifizierten GPR Environments mit der ursprünglichen Version.

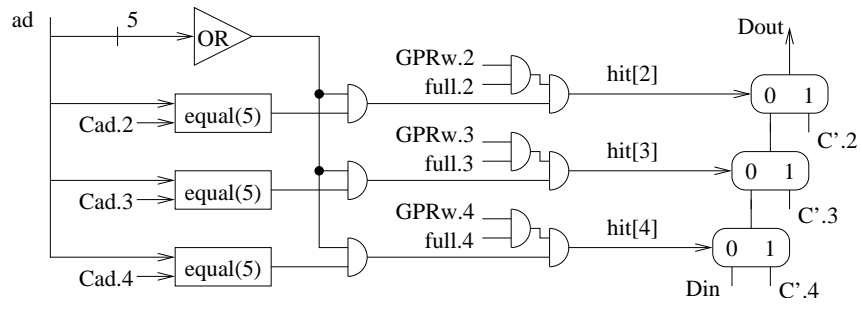


Figure 1: Forwarding Schaltkreis aus der Vorlesung

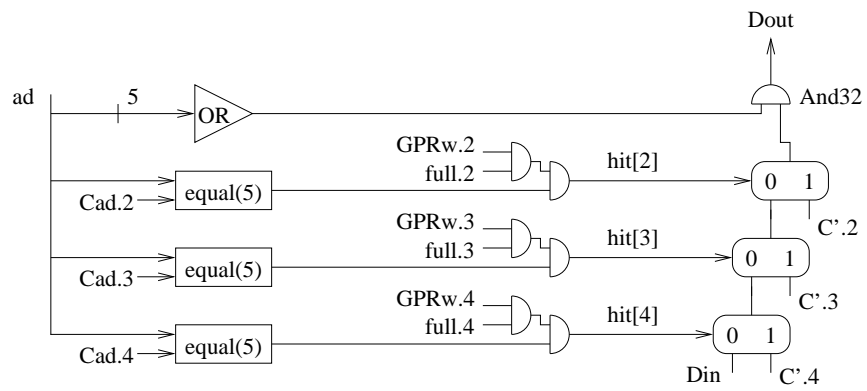


Figure 2: Modifizierter Forwarding Schaltkreis