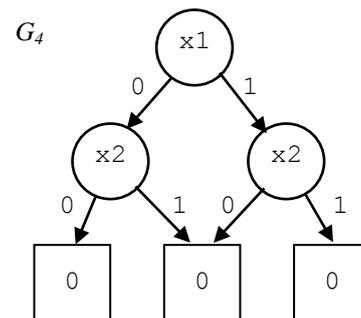
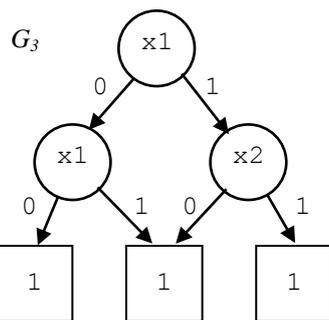
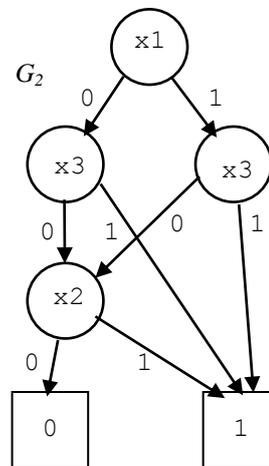
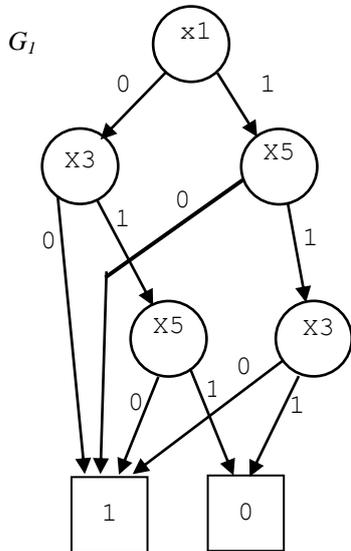


**Rechnerstrukturen im WS 2011/2012**  
**Übungsblatt 3**

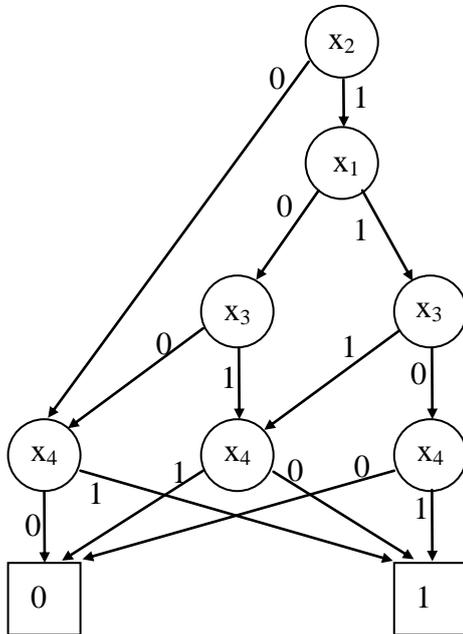
**Aufgabe 1 (OBDDs) (4 Punkte)**

Gegeben seien vier Graphen  $G_1, G_2, G_3, G_4$ , die Funktionen  $f_1, f_2, f_3, f_4 : B^3 \rightarrow B$  repräsentieren sollen. Entscheiden Sie, ob diese  $\pi$ OBDDs darstellen. Begründen Sie Ihre Antwort. Falls ein Graph ein  $\pi$ OBDD ist, reduzieren Sie ihn schrittweise unter Angabe der angewendeten Regel.



**Aufgabe 2 (OBDDs) (4 Punkte)**

Sie sollen das unten angegebene  $\pi$ OBDD schrittweise unter Anwendung der Reduktionsregeln reduzieren. In jedem Schritt darf jeweils nur eine Reduktionsregel einmal angewendet werden. Geben Sie in der nummerierten Liste jeweils die angewendete Regel an und markieren Sie die Knoten, auf welche die Regel angewendet werden soll, indem Sie die entsprechende Zeilennummer in die betroffenen Knoten schreiben. Sie brauchen nicht nach jeder Regelanwendung ein neues  $\pi$ OBDD zeichnen. Nur das reduzierte  $\pi$ OBDD stellen Sie unten einmal gesondert dar. Es müssen nicht so viele Regelanwendungen gefunden werden, wie Zeilen in der Liste zur Verfügung stehen.



1. \_\_\_\_\_
2. \_\_\_\_\_
3. \_\_\_\_\_
4. \_\_\_\_\_
5. \_\_\_\_\_

**Aufgabe 3 (Rechnen mit Zweierkomplementen) (4 Punkte)**

Berechnen Sie:

- 1) 10110011 + 01111010
- 2) 00100011 - 01000010
- 3) 00010110 - 10011101
- 4) 01101111 + 00100001

Die Zahlen sind in Zweierkomplementdarstellung (8-bit Breite) gegeben. Geben Sie das Ergebnis ebenfalls in dieser Darstellung an. Geben Sie an, wenn ein Ergebnis ungültig ist.

**Aufgabe 4** (Additionsschaltnetze) (4 Punkte)

Es sollen zwei 8bit binär codierte Betragszahlen  $x = (x_7 \dots x_0)_2$  und  $y = (y_7 \dots y_0)_2$  addiert werden. Wie Sie wissen, werden dafür Überträge  $ü_i$  berechnet. Dabei entsteht der Übertrag  $ü_i$  bei der Addition von  $x_{i-1}$ ,  $y_{i-1}$  und  $ü_{i-1}$ . Uns interessiert die Laufzeit, bis das Ergebnis der höchsten Stelle  $c_8$  vorliegt. ( $c_8$  wird an den Eingang der nächsten Stelle weitergeleitet)

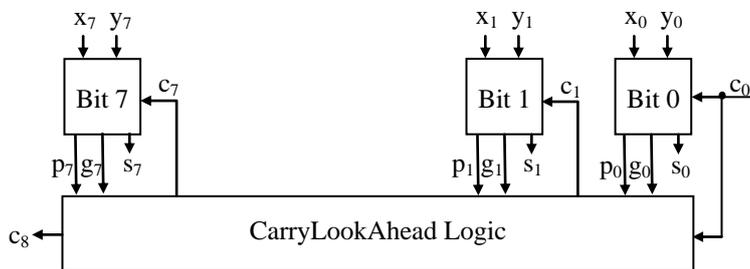
1. Skizzieren Sie das Addierwerk für einen Ripple-Carry-Adder mit 8 bit Breite. Verwenden Sie für die Skizze möglichst wenige Halb- und Volladdierer. Skizzieren Sie ebenfalls den Aufbau eines Halb- und Volladdierers. Es stehen die Gatter „und“, „oder“ und „xor“ zur Verfügung. Die Laufzeit eines Gatters sei jeweils  $t$ . Wie groß ist die Laufzeit  $t_g$  bis das Ergebnis von  $c_8$  vorliegt? Die Laufzeit der Halb- und Volladdierer hängt von deren innerem Aufbau ab und ist entsprechend zu berücksichtigen.
2. Betrachten Sie die Laufzeit eines Carry-Look-Ahead-Addierers. Für die  $g_i$  und  $p_i$  in den Blackboxes „Bit  $i$ “ benötigen Sie jeweils einen Halbaddierer. Die Berechnung der Summen  $s_i$  wird hier nicht betrachtet. Den Inhalt der CarryLookAhead Logic sollen Sie selber erforschen.

Wie wird der Übertrag  $c_8$  berechnet? (Formel)

Wie groß ist die Tiefe der CarryLookAhead Logic  $c_8$ ?

Hinweis: Sie können Gatter mit beliebig großem Fan-In verwenden.

Wie groß ist die Gesamtlaufzeit  $t_g$  bis das Ergebnis von  $c_8$  vorliegt? (jedes verwendete Gatter habe eine Laufzeit von  $t$ )



**Die Abgaben sollen bis Mittwoch, den 02. November 2011 um 18.00 Uhr in die Briefkästen im Pavillon 6 eingeworfen werden. Bitte Name (bei einem 3er-Team alle), Matrikel- und Gruppennummer oben auf der ersten Seite der Lösungen angeben.**