



Abb. 3.4: Die verfeinerte Version der Multiplikationshardware. Vergleichen Sie diese mit der ersten Version in Abbildung 3.2. Das Multiplikandenregister, die ALU und das Multiplikatorregister sind alle 32 Bit breit. Nur das Produktregister weist noch 64 Bit auf. Nun wird das Produkt nach rechts verschoben. Das getrennte Multiplikatorregister ist verschwunden. Stattdessen befindet sich der Multiplikator nun in der rechten Hälfte des Produktregisters. Diese Änderungen sind grau hervorgehoben. (Das Produktregister sollte eigentlich 65 Bit breit sein, um den Übertrag aus dem Addierer aufzunehmen. Hier ist es mit 64 Bit dargestellt, um die Weiterentwicklung aus Abbildung 3.2 zu verdeutlichen.)

Auch bei der Multiplikation von Konstanten kann Arithmetik durch Schiebeoperationen ersetzt werden. Einige Compiler ersetzen Multiplikationen mit kurzen Konstanten durch mehrere Schiebeoperationen und Additionen. Da eine Verschiebung um ein Bit nach links bei Zahlen zur Basis 2 deren Wert verdoppelt, bewirkt das Verschieben der Bits nach links dasselbe wie die Multiplikation mit einer Potenz von 2. Wie in Kapitel 2 bereits erwähnt, führt nahezu jeder Compiler die Optimierung *Strength Reduction* durch, bei der eine Multiplikation mit einer Potenz von 2 durch eine Schiebeoperation nach links ersetzt wird.

Hardware/
Software-
Schnittstelle

Ein Multiplikationsalgorithmus

Nehmen Sie 4-Bit-Zahlen, um Platz zu sparen, und multiplizieren Sie $2_D \times 3_D$, d. h. $0010_B \times 0011_B$.

Tabelle ?? führt die Werte der einzelnen Register für jeden der Schritte auf, wobei deren Nummerierung der in Abbildung 3.3 folgt. Der letzte Wert ist $0000\ 0110_B$, also 6_D . Die Registerwerte, die sich beim jeweiligen Schritt ändern, sind grau hinterlegt. Das jeweils unterstrichene Bit ist das Bit, das geprüft wird, um zu bestimmen, welche Operation im nächsten Schritt auszuführen ist.

Beispiel

Lösung

3.3.2 Multiplikation mit Vorzeichen

Bisher haben wir positive Zahlen betrachtet. Der Umgang mit vorzeichenbehafteten Zahlen ist am einfachsten, wenn zunächst der Multiplikator und der Multiplikand in positive Zahlen umgewandelt und die ursprünglichen Vorzeichen gemerkt werden. Der Algorithmus muss 31 Iterationen durchlaufen, wobei die Vorzeichen bei der Berechnung nicht berücksichtigt werden. Wie wir

Das verwendete Tabellenprogramm ist nicht fuer farbige oder graue Zellen ausgelegt!