

Low-Power Multipliers with Data Wordlength Reduction

Kyungtae Han, Brian L. Evans, and Earl E. Swartzlander, Jr.

Dept. of Electrical and Computer Engineering

The University of Texas at Austin

Austin, TX 78712–1084 USA

Email: khan@mail.utexas.edu, bevans@ece.utexas.edu, eswartzla@aol.com

Abstract— Multiprecision multipliers reduce power consumption by selecting smaller multipliers (i.e., submultiplier) according to the wordsize of the input operands. However, arbitrary levels of bit precision are not achieved by multiprecision multipliers. Two proposed wordlength reduction techniques that reduce power consumption with arbitrary levels of bit precision are considered. Expectation values of bit switching activity for reduction in the signed right shift method and the truncation method are derived. The signed right shift method and the truncation method are applied to a 16-bit radix-4 modified Booth multiplier and a 16-bit Wallace multiplier. The truncation method with 8-bit operands reduces the power consumption by 56% in the Wallace multiplier and 31% in the Booth multiplier. The signed right shift method shows no power reduction in the Wallace multiplier and 25% power reduction in the Booth multiplier. Unequal levels of precision in operands show different power reduction value for the Booth multiplier. The non-recoded operand in the Booth multiplier with 8-bit reduction has 13% more sensitivity in power consumption than the recoded multiplicand.

I. INTRODUCTION

Computing systems demand minimizing the power dissipation due to limited battery power in portable computing and the difficulty of cooling in high speed signal processing. Many methods have been developed to reduce power consumption. Lowering the supply voltage and minimizing the hardware are used for low-power hardware [1]. Changing the instruction order and reducing the number of operations are used for low-power software [2]. A major focus of low power design is to reduce the switching activity to the minimal level required to perform the computation, since to a first order the power consumption of CMOS circuits is proportional to the number of gate transitions [3].

Multipliers are usually a major source of power consumption in typical DSP applications. Multiprecision multipliers have been developed for low-power consumption [4], [5]. In multiprecision multipliers, multiplications are performed by 8-bit, 16-bit or 24-bit circuits according to the input operand size. Power reduction of up to 66% is achieved in [4] and 58% in [5]. However, arbitrary operand sizes such as 10-bits are not accommodated efficiently in these approaches. A wordlength reduction technique has been proposed in [6] to select any word size. The wordlength reduction technique shows a 72% reduction of average gate transitions. An extension of the wordlength reduction technique is presented in this paper.

Overviews of wordlength reduction techniques and power reduction methods are presented in Sections II and III, respectively. Expectation values of bit switching in inputs are derived in Section IV. A radix-4 modified Booth multiplier and a Wallace multiplier, which are used in simulations are explained Section V. Power consumption in these multipliers is estimated for FPGA implementations in Section VI. Also the power consumption of multipliers where the operands are of different sizes is estimated and compared.

II. WORDLENGTH REDUCTION IN MULTIPLIERS

Previous multiprecision multipliers have a few choices of operand precision due to hardware limitations [4], [5]. The multiprecision multiplier does not accommodate arbitrary precision due to its fixed hardware structure. For example, with 10-bit operands, a multiprecision multiplier, which supports 8-bit and 16-bit multiplication, has to use 16-bit multiplication with 6 unnecessary bits. Data wordlength reduction techniques can reduce the unnecessary switching activity.

There are two kinds of data wordlength reduction. One is reduction via right-shifting, while the other is reduction via left-shifting i.e., with truncation. The right-shifting method moves data from the most significant (MS) side to least significant (LS) side with sign extension. The sign extension bits are all ones when the operand is negative and all zeros when the operand is positive. The truncation method removes data from the LS side. An example of 8-bit reduction from 16-bit multiplication is shown in Figure 1. The original 16-bit multiplication is shown in Figure 1(a). The reduction by an 8-bit right-shift moves 8 bits data in the MS side to the LS side with sign extension as shown in Figure 1(b). The signed right shifted value becomes '1111 1111 1101 1100', because the original value, '1101 1100 1010 1001', is negative. The reduction by 8-bit truncation removes 8-bit data in the LS side by masking the input data with '1111 1111 0000 0000' with the result that is shown in Figure 1(c).

III. POWER REDUCTION VIA WORDLENGTH REDUCTION

Power dissipation in digital CMOS circuits can be classified as switching power consumption and static power consumption. The switching power is proportional to the switching activity parameter, α in

$$P_{switching} = \alpha C_L V_{dd}^2 f_{clk} \quad (1)$$

$$\begin{array}{r}
0001\ 0010\ 0011\ 0100 \\
\times\ 1101\ 1100\ 1010\ 1001 \\
\hline
\end{array}$$

(a) Original multiplication

$$\begin{array}{r}
0001\ 0010\ 0000\ 0000 \\
\times\ 1101\ 1100\ 0000\ 0000 \\
\hline
\end{array}$$

(b) Reduction by truncation

$$\begin{array}{r}
0000\ 0000\ 0001\ 0010 \\
\times\ 1111\ 1111\ 1101\ 1100 \\
\hline
\end{array}$$

(c) Reduction by signed right shift

Fig. 1. Example of 8-bit data wordlength reduction

Where: C_L is the load capacitance, V_{dd} is the operating voltage, and f_{clk} is the operating frequency [3]. The term αC_L can be viewed as the effective switching capacitance of the transistor nodes. Therefore, minimizing switching activity can effectively reduce the power dissipation without impacting the circuit performance [7]. Wordlength reduction methods in Section II can minimize switching activity at the expense of data precision as in [6]. The minimized switching activity reduces power consumption as shown in Eq. (1).

The wordlength reduction methods can be applied to low-power instruction based processors or FPGA/reconfigurable hardware. The truncation method is implemented by adding mask modules, which consist of N-bit AND gates, in front of the multiplier inputs. The signed right shift method uses shift registers and sign extension units. Therefore, the truncation method needs less extra hardware than the signed right shift method for its implementation.

IV. EXPECTATION OF SWITCHING

Power consumption in CMOS digital circuits is proportional to switching activity in logic gates. Logic gates in multipliers are switched after input multiplicand data are changed from previous data. The total number of gates that switch is used to calculate switching power consumption. It is hard to predict the overall number of gates that switch in a multiplier due to the glitch effect, which unexpectedly increases the switching activity.

Multiplicand inputs propagate the switching activity into inner logic gates in a combinational multiplier. The expected value of input switching is a meaningful factor to predict the number of gates that switch in a multiplier. In this section the expected value of the number of gates that switch in L-bit

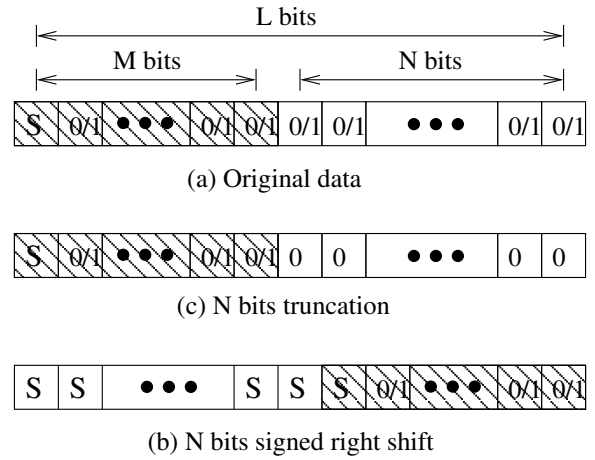


Fig. 2. Bit operation in effective bits, M . S is a signed bit

inputs and M-bit reduction by truncation or signed right shift methods is estimated.

A. L-bit input

Let X be a random variable of the number of total bits switched in wordlength L as in Fig. 2. Each bit in the data has equal probability of bit switching such as zero to one or one to zero, when new input data are given in previous data locations. The probability of the switching of each bit is $\frac{1}{2}$. The switching probability in X has binomial distribution:

$$P_X(x) = \binom{L}{x} \left(\frac{1}{2}\right)^x \left(\frac{1}{2}\right)^{L-x} \quad (2)$$

The expected value of X is

$$E(X) = \sum_{x=0}^L x \cdot P_X(x) \quad (3)$$

The expected value of a binomial distribution with probability, p , and the number of trials, l , is $l \cdot p$. The expected value of switching in L bits can be simplified to

$$\begin{aligned}
E(X) &= L \cdot p \\
&= \frac{L}{2}.
\end{aligned} \quad (4)$$

The expected value of switching in L bits is half of L bits.

B. N-bit truncated data in L-bit input

The effective bit-width can be reduced by truncation. When truncated data are consecutively used as input data, only the remaining bits have probability of switching as shown in Fig. 2(b). N -bit truncated data in L bit width input have $L - N$ effective width to be switched, while N bits have always zero values. The expectation of N -bit truncated data in L bit inputs is

$$E_{tr}(X) = \frac{L - N}{2} \quad (6)$$

$$= \frac{M}{2} \quad (7)$$

where M is the number of bits that are not truncated. These equations show that the expectation value of switching in truncated data is half of the remaining data width.

C. Signed right shift

The effective bit-width can be reduced by right shifting. The signed right shift moves data to right side with the sign bit filled into the vacated bit positions. N bit signed right shifted data in L bit input add N additional sign bits as shown in Fig. 2 (c). The expected value of switching in N bit signed right-shifted data can be obtained using a conditional expectation [8] with a random variable, Y , of a sign bit switching as

$$E_{rs}(X) = E(E(X|Y)) \quad (8)$$

$$= \sum_{s=0}^1 P(Y = s)E(X|Y = s) \quad (9)$$

$$= \frac{1}{2}E(X|Y = 0) + \frac{1}{2}E(X|Y = 1) \quad (10)$$

Where: s is the sign bit. The first term in the right side in (10) gives the expected value when the sign bit is not changed. Thus, only $M - 1$ bits change. From Eqs. (2), (3), and (5), the first term of conditional expectation value (10) becomes

$$\begin{aligned} E(X|Y = 0) &= \sum_{x=0}^{M-1} x \cdot \binom{M-1}{x} \left(\frac{1}{2}\right)^x \left(\frac{1}{2}\right)^{M-1-x} \quad (11) \\ &= \frac{M-1}{2} \quad (12) \end{aligned}$$

where $M = L - N$.

The second term in the right side in (10) is the conditional expectation when the sign bit is switched. The N bit signed right shifted data have $N + 1$ sign bits as shown in Fig. 2 (c). The conditional expectation of switched sign bit, $E(X|Y = 1)$, is

$$\sum_{x=0}^{M-1} (x + N + 1) \binom{M-1}{x} \left(\frac{1}{2}\right)^x \left(\frac{1}{2}\right)^{M-1-x} \quad (13)$$

The x in the summation in Eq. (13) can be separated as

$$\frac{M-1}{2} + \sum_{x=0}^{M-1} (N+1) \binom{M-1}{x} \left(\frac{1}{2}\right)^x \left(\frac{1}{2}\right)^{M-1-x} \quad (14)$$

$$= \frac{M-1}{2} + \sum_{x=0}^{M-1} (N+1) \binom{M-1}{x} \left(\frac{1}{2}\right)^{M-1} \quad (15)$$

In general, the sum of all the combinations of K distinct things gives

$$\sum_{x=0}^K \binom{K}{x} = 2^K \quad (16)$$

Using Eqs. (16) and (15) yields the conditional expectation value as

$$E(X|Y = 1) = \frac{M-1}{2} + (N+1) \left(\frac{1}{2}\right)^{M-1} 2^{M-1} \quad (17)$$

$$= \frac{M}{2} + N + \frac{1}{2} \quad (18)$$

TABLE I
EXPECTATION OF SWITCHING IN L BIT INPUT

Inputs	Expectation of switching
Full length used	$L/2$
N bit truncation	$M/2$
N bit signed right shift	$L/2$

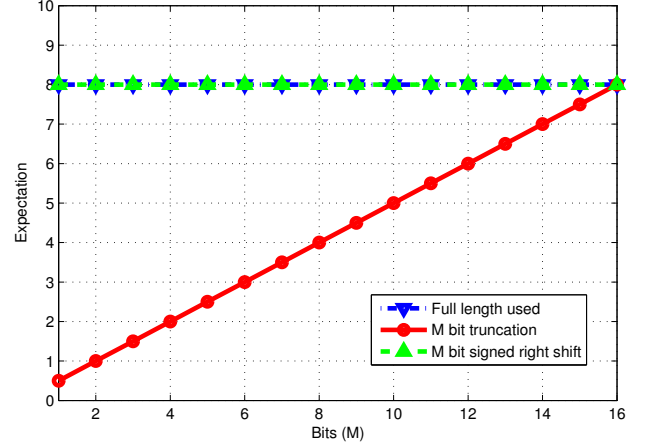


Fig. 3. Expectation of number of switching in inputs

From Eqs. (12) and (18), the expectation of switching data in (10) can be simplified to

$$E_{rs}(X) = \frac{1}{2} \left(\frac{M-1}{2}\right) + \frac{1}{2} \left(\frac{N}{2} + N + \frac{1}{2}\right) \quad (19)$$

$$= \frac{M+N}{2} \quad (20)$$

$$= \frac{L}{2} \quad (21)$$

The expected value of the number of bits switched in N -bit signed right shift data in L bits input is half of L regardless of the signed right shift. Therefore, the expected value of switching in signed right shifted input is the same as for an unshifted input. The expected values are summarized in Table I and are shown in Fig. 3.

V. MULTIPLIER

The hardware multiplier on most Programmable DSPs uses either a Wallace multiplier or a Radix-4 modified Booth multiplier [9]. For example, the TI TMS320C64 uses the Wallace algorithm and the TI TMS320C62 uses the Radix-4 modified Booth algorithm.

A. Wallace Multiplier

In a tree-based multiplier, partial products are added using full adders or half adders. In 1964, Wallace showed a tree structure, which is an efficient method to add partial products [10]. A Dadda dot diagram of a 4-bit Wallace multiplier is shown in Figure 4. Rows are grouped into sets of three during each reduction stage. Within each three row set, (3,2) counters

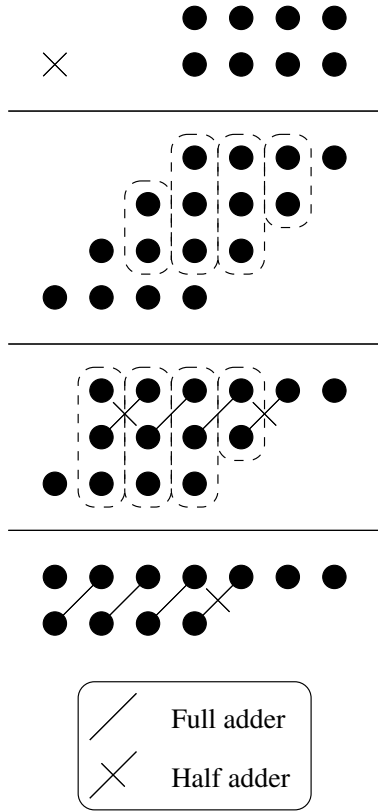


Fig. 4. Dadda dot diagram for 4-bit Wallace multiplier

reduce columns with three bits to two bits and (2,2) counters reduce columns with only two bits. Rows that are not part of a three row set are transferred to the next stage without modification. [11].

B. Radix-4 Modified Booth Multiplier

Booth recoding is a commonly used technique to recode one of the operands in binary multiplication. Fig. 5 shows a radix-4 modified Booth multiplier of $a \times x$. A two's complement multiplier, x , is recoded as a radix-4 number, z , which dictates the multiples $-2a$, $-a$, 0 , a , and $2a$ to be added to the cumulative partial product. The radix-4 Booth's recoding is shown in Table II.

VI. SIMULATION RESULTS AND DISCUSSIONS

A 16-bit Wallace Multiplier and a 16-bit Radix-4 modified Booth multiplier are used for power estimation with data wordlength reduction. The multipliers are synthesized for Xilinx, XC3S200-5FT256 FPGA [12]. The XPower tool estimates the power consumption of this FPGA with different operand sizes. The dynamic power is estimated across VCCINT, which is a power supply pin of the dedicated internal core with 1.2 V supply. The operational frequency of the multipliers is set to 1 MHz.

Power estimates for a 16-bit Wallace multiplier are shown in Figure 6. An average power of 0.45 mW is consumed with 16-bit data operands in the Wallace multiplier. As the

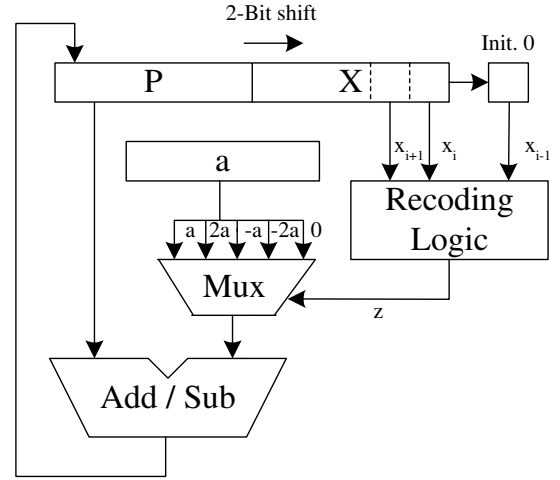


Fig. 5. A Radix-4 multiplier based on Booth's recoding. The a and x are multiplicands. P is product of multiplication. Three bits in X are recoded to z .

TABLE II
RADIX-4 BOOTH'S RECODING. THE A AND X ARE MULTIPLICANDS. 3 BITS OF X ARE RECODED INTO Z .

x_{i+1}	x_i	x_{i-1}	z	action
0	0	0	0	0
0	0	1	1	a
0	1	0	1	a
0	1	1	2	$2a$
1	0	0	-2	$-2a$
1	0	1	-1	$-a$
1	1	0	-1	$-a$
1	1	1	0	0

operand size is reduced, the truncation method decreases the power consumption. The average power reduction in 8 bit wordlength reduction by the truncation method is 56%. The right shift method shows little or no power reduction due to the sign extension. The extended sign bits are added to the input whenever a right shift occurs. These bits affect the switching activity. Therefore, the signed right shift method is not recommended for low-power Wallace multipliers.

Power estimates for a 16-bit radix-4 modified Booth multiplier are shown in Figure 7. A power of 0.52 mW is consumed with 16-bit operands in the Booth multiplier. As the data wordlength is reduced by either the truncation method or the signed right shift method, the average power consumption decreases. The average power consumption for multipliers with 8 bits operands implemented by the signed right shift and the truncation methods are decreased by 25% and 31%, respectively.

The power consumption for the Wallace multiplier as shown in Figure 6 shows a trend that matches the expectations from Figure 3. The amount of switching is not changed in signed

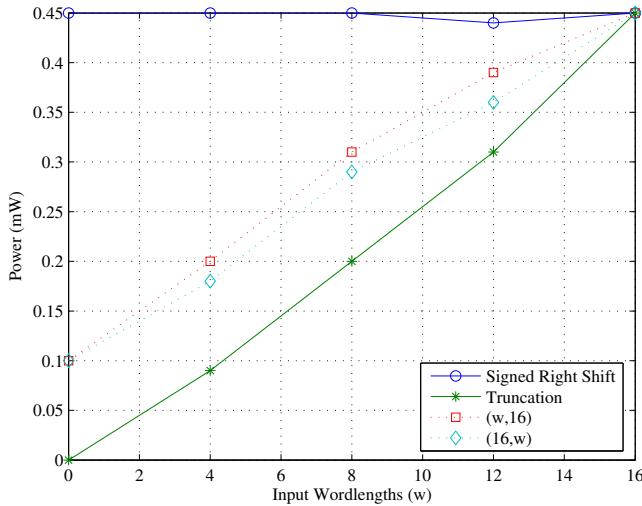


Fig. 6. Dynamic power consumption in 16-bit \times 16-bit Wallace multiplier (1MHz)

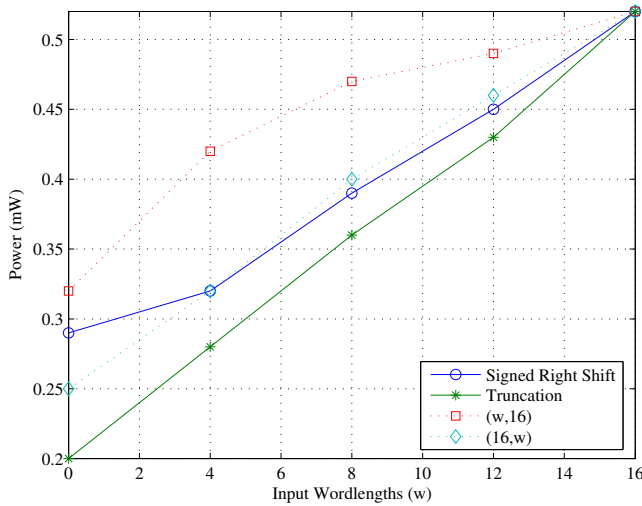


Fig. 7. Dynamic power consumption in 16-bit \times 16-bit Radix-4 modified Booth multiplier (1MHz)

right shift input, but it is changed in truncated input as the input effective wordlength changes. However, in the Booth multiplier, the power consumption of signed right shift input as shown in Figure 7 is changed as the input effective wordlength changes.

The average power consumption is also estimated when operands have unequal sizes. One of the operands is reduced with the truncation method while the other operand is fixed at 16 bits. The first and the second element in the parentheses in Figure 6 and Figure 7 represent two multiplicands in multiplication. When operands are swapped such as (A by X) to (X by A), the power consumption shows different results. For the Wallace multiplier there is a small difference, but the Booth multiplier has a large difference because of its

asymmetric structure. The first and the second operand in the Booth multiplier represent a recoded input, X , and a non-recoded input, A , respectively as shown in Fig. 5. The result shows that when the non-recoded input's level of precision is reduced, the average power decreases by 13% more than when the recoded input is reduced for 8-bit wordlength reduction. The reason is that the non-recoded input, which is routed to multiplexers and to adder/subtractor logic, affects more power consumption than the recoded input. Therefore, in the Booth multiplier, data wordlength reduction in the non-recoded operand achieves more power reduction than that in the recoded operand.

VII. CONCLUSION

Two kinds of input data wordlength reduction methods in multipliers have been examined and analyzed for low power consumption. A truncation method with 8 bits reduces power consumption by 56% in a 16-bit Wallace multiplier and 31% in a 16-bit radix-4 modified Booth multiplier. A signed right shift method exhibits no power reduction in the Wallace multiplier and 25% reduction in the Booth multiplier. When the operands have different sizes, the multipliers also show power reduction. In particular, the non-recoded operand in the Booth multiplier is 13% more sensitive in power consumption than the recoded multiplicand. This difference can be exploited in a low-power digital filter design with low-precision coefficients.

REFERENCES

- [1] K. K. Parhi, *VLSI Digital Signal Processing Systems*. New York, NY: John Wiley & Sons, 1999.
- [2] M. T. Lee, V. Tiwari, S. Malik, and M. Fujita, "Power analysis and minimization techniques for embedded DSP software," *IEEE Trans. on VLSI Systems*, vol. 5, pp. 123–135, Mar. 1997.
- [3] A. P. Chandrakasan and R. W. Brodersen, "Minimizing power consumption in digital CMOS circuits," *Proc. IEEE*, vol. 83, pp. 498–523, Apr. 1995.
- [4] J. Y. F. Tong, D. Nagle, and R. A. Rutenbar, "Reducing power by optimizing the necessary precision/range of floating-point arithmetic," *IEEE Trans. VLSI Syst.*, vol. 8, pp. 273–285, June 2000.
- [5] H. Lee, "A power-aware scalable pipelined Booth multiplier," in *Proc. IEEE International Systems-On-Chip Conference*, Sept. 2004, pp. 123–126.
- [6] K. Han, B. L. Evans, and E.E. Swartzlander, Jr., "Data wordlength reduction for low-power signal processing software," in *Proc. IEEE International Workshop on Signal Processing Systems*, Austin, TX, Oct. 2004, pp. 343–348.
- [7] O. T.-C. Chen, S. Wang, and Y.-W. Wu, "Minimization of switching activities of partial products for designing low-power multipliers," *IEEE Trans. on VLSI Systems*, vol. 11, pp. 418–433, June 2003.
- [8] G. Grimmett and D. Stirzaker, *Probability and Random Processes*. Oxford University Press, 2001.
- [9] B. Parhami, *Computer Arithmetic algorithm and hardware designs*. Oxford University Press, 2000.
- [10] C. S. Wallace, "A suggestion for a fast multiplier," *IEEE Trans.. on Computers*, vol. 13, pp. 14–17, 1964.
- [11] K. C. Bickerstaff, E.E. Swartzlander, Jr., and M. J. Schulte, "Analysis of column compression multipliers," in *Proc. IEEE Symposium on Computer Arithmetic*, June 2001, pp. 33–39.
- [12] *Spartan-3 FPGA Family: Complete Data Sheet*, Xilinx, Jan. 2005. [Online]. Available: <http://www.xilinx.com/bvdocs/publications/ds099.pdf>