Table 13.5 Truth Table for Low-Power Booth Recoding Circuitry in Problem 12

y_{2j+1}	y_{2j}	y_{2j-1}	y'_j	neg	$x1$	$x2$	zp
0	0	0	0	0	0	1	1
0	0	1	1	0	1	0	*
0	1	0	1	0	1	0	*
0	1	1	2	0	0	1	0
1	0	0	−2	1	0	1	0
1	0	1	−1	1	1	0	*
1	1	0	−1	1	1	0	*
1	1	1	0	1	0	1	1

(b) Draw the complete modified Booth recoded bit-serial multiplier architecture and show all switching instances.

12. This problem considers design of a Booth recoding circuit for low power. The control circuitry for Booth recoding in Problem 11 contains only two XOR gates and one inverter. However, the generation of the control signals A_i, B_i, and C_i through different logic levels with different propagation delay time leads to an increase in the glitching activity in the partial product generation and accumulation circuit, hence leading to high power dissipation. Glitching and power consumption can be reduced by using a redundant recoding scheme, which balances the paths from each input to the outputs of the control circuitry [11].

Consider the multiplication $A \times Y$, and denote the recoded multiplier bits as y'_j. Four control signals are used in the new *race-free* Booth recoder, neg, $x1$, $x2$ and zp. $neg = 1$ implies y'_j is negative, and $neg = 0$ implies y'_j is nonnegative. $x1 = 1$ implies $|y'_j| = 1$; $x2 = \overline{x1}$. zp distinguishes $|y'_j| = 2$ from $|y'_j| = 0$. The new recoding scheme is summarized in the Table 13.5, where * denotes a *don't-care* condition. One bit-slice of the partial product generator obtained using this recoding scheme is shown in Fig. 13.41.

(a) Prove that the circuit in Fig. 13.41 generates the correct partial products for the Booth multiplication.

(b) Derive the Booth recoder (control) circuit such that the propagation delay of all paths from each input bit, including y_{2j+1}, y_{2j}, y_{2j-1}, a_i and a_{i-1}, to each bit of the partial product, Ay'_j, are equal and are limited by one XOR/XNOR gate delay, one AND gate delay, and one NOR gate delay. The *don't care* conditions in zp signal can be used for logic minimization. Show those zp values

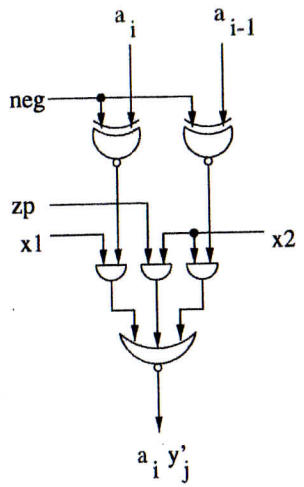


Fig. 13.41 One bit slice partial product generator for the *race-free* Booth multiplier in Problem 12.

13. Obtain a 2-unfolded carry-ripple digit-serial multiplier by unfolding the Lyon's multiplier in Fig. 13.17.

14. This problem is concerned with fixed-point bit-serial and digit-serial implementation of an all-pole second-order recursive filter

$$y(n) = -\frac{7}{8}y(n-1) + \frac{3}{4}y(n-2) + x(n).$$

Assume the signal wordlength to be 8 (i.e., the wordlengths for y and x are 8 bits). Assume the coefficient $-7/8$ is encoded as 1.001 and 3/4 is encoded as 0.11. In the multiplication with respect to coefficients, only multiplication with respect to nonzero bits is carried out.

(a) Draw the block diagram of the computation by exploiting associativity. The inner loop bound of your architecture should be limited by 1 multiply-add time.

(b) Design a functionally correct bit-level pipelined bit-serial architecture for the structure in part (a).

(c) The loop latency in part (b) imposes a constraint on the minimum wordlength for the signals y and x. What is the minimum feasible signal wordlength for the system in part (b)?

15. This problem considers bit-serial implementation of the 3rd-order fixed-coefficient IIR filter shown in Fig. 13.42. Assume the data wordlength to

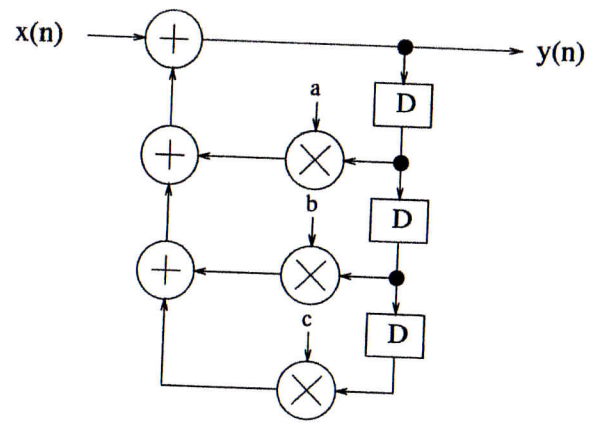


Fig. 13.42 IIR filter diagram for Problem 15.

$b = 1/8$, $c = -3/4$. Assume all numbers to be represented using two's complement representation.

(a) Obtain an equivalent structure by using associativity such that the inner loop bound is limited by 1 multiply-add time.

(b) Complete the bit-level pipelined bit-serial design of the structure in part (a). Treat minimization of delay elements in the design as a secondary objective.

16. This problem addresses a bit-level pipelined bit-serial implementation of the recursive computation

$$y(n) = \frac{3}{16}y(n-1) + \frac{5}{8}y(n-2) + \frac{1}{2}x(n)$$

using two's complement representation and a signal wordlength of 10. The bit-serial design must use Horner's rule to improve accuracy in the precision available. Associativity must be exploited so that the loop bound of the inner loop is limited by 1 multiply and 1 add time at the word level. Design the bit-serial architecture for a signal wordlength of 10. What is the minimum feasible wordlength for this computation?

17. Consider the bit-serial implementation of the computation

$$y(n) = 0.25y(n-1) + x(n) \tag{13.40}$$

shown in Fig. 13.43.

What wordlength has been used in the circuit? For this wordlength, complete the missing switching instances. Unfold this bit-serial structure by a factor of 3 to obtain a digit-serial implementation with digit size 3.

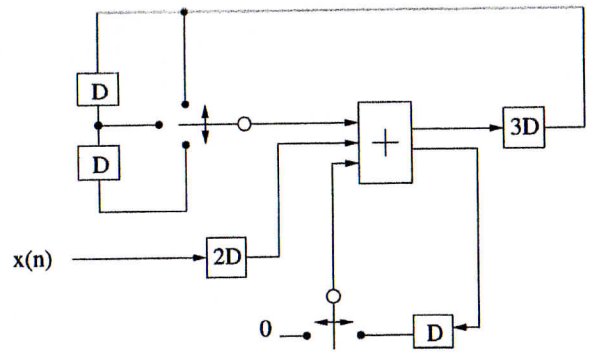


Fig. 13.43 Bit-serial implementation of the IIR filter in Problem 17.

18. This problem considers the design of a low-latency bit-serial multiplier by transforming the ripple-carry Lyon's multiplier using associativity and retiming.

(a) Show that the architecture in Fig. 13.44(a) can be transformed to that in Fig. 13.44(b) using associativity and retiming. Compare the critical path and latencies of these 2 architectures.

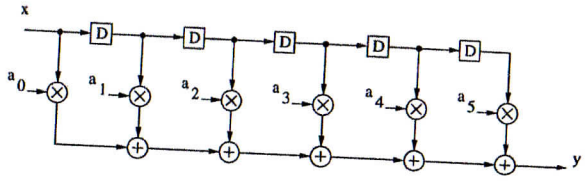
(b) Design a low-latency bit-serial multiplier by applying similar transformations as in part (a) to the Lyon's ripple-carry bit-serial multiplier or by recasting the Horner's rule multiplication equation into an appropriate form. Assume a wordlength of 12 for signal sample and the coefficient. Show all switching instances in your architecture.

(c) Design a low latency constant bit-serial multiplier to compute $y(n) = ax(n)$, where $a = 0.01011100101$, by considering multiplication with respect to nonzero bits only using the architecture in part (b). Show all the switching instances in this architecture. Assume a signal wordlength of 12.

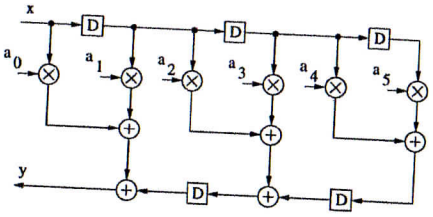
(d) Based on the bit-serial multiplier derived in part (c), design a bit-serial implementation of the recursive computation $y(n) = ay(n-1)+x(n)$ for the same value of a in (c). Assume a signal wordlength of 12.

19. Obtain the CSD representation of the following two's complement numbers:

$$c_0 = 0.00010110, \quad c_1 = 0.01001100, \quad c_2 = 0.01100110,$$
$$c_3 = 0.00010110111.$$



(a)



(b)

Fig. 13.44 Bit-serial multipliers in Problem 18.

20. Obtain the CSD representation of the following two's complement numbers:

$$c_0 = 1.11111111, \quad c_1 = 1.11110010, \quad c_2 = 0.01100110,$$
$$c_3 = 0.01001100111.$$

21. This problem considers a bit-serial implementation of the IIR filter

$$y(n) = a_1 y(n-1) + a_2 y(n-2) + x(n), \tag{13.41}$$

where $a_1 = 0.00011011100$ and $a_2 = 0.01001111110$.

(a) Represent the coefficients a_1 and a_2 in CSD representation.

(b) Using CSD representation and the data-flow graph in Fig. 13.27(b), obtain a bit-level pipelined bit-serial IIR filter architecture for a signal wordlength of 12. What is the minimum feasible wordlength of this architecture?

(c) Apply the Horner's rule and tree height reduction techniques to improve your design in part (b). What is the minimum feasible wordlength of this design?

22. Extend the OBC concept to implement a distributed arithmetic architecture using a ROM containing $2^N/4$ words.

23. Consider ROM decomposition with N=16, K=4. Calculate the reduc
tion factor in ROM

24. This problem considers design of digit-serial distributed arithmet
chitecture. Implement the computation

$$Y = \sum_{i=0}^{N-1} c_i x_i \tag{(
$$

using a 3-bit digit-serial distributed arithmetic assuming a wordl
of 9. The bits $x_{i,3k+l}$ should be processed in the l-th ROM (l = 0
in k-th clock cycle ($k = 0, 1, 2$). Using 3 ROMs and a multi-input
accumulator, design the 3-bit digit-serial distributed arithmetic
tecture. This architecture processes 3 consecutive input bits in a
cycle and all input words are processed in 3 clock cycles. Assum
ROM contents to be represented using 16 bits. Design two for
shift-accumulators using bit-parallel (i) carry-ripple and (ii) carry
adders. The output of the shift-accumulator is represented usi
bits. The critical path of the carry-ripple design should be $17t_{FA}$
that of the carry-save design should be $3t_{FA}$, where t_{FA} is the pro
tion delay of a full adder. The carry-save design requires an addi
carry-propagate adder at the end.

(a) Calculate the latencies of both designs in terms of t_{FA}.

(b) Which design is more suitable for high speed? Why?

(c) Which design is more suitable for low power? Why?

REFERENCES

1. L. B. Jackson, J. F. Kaiser, and H. S. McDonald, "An approach to i
mentation of digital filters," *IEEE Trans. on Audio Electroacoustics*, v
pp. 413–421, 1968.

2. P. B. Denyer and D. Renshaw, *VLSI Signal Processing: A Bit-Serial App*
Addison-Wesley, 1986.

3. R. Jain et al., "Custom design of a VLSI PCM-FDM transmultiplexor
system specification to circuit layout using a computer aided design sys
IEEE J. Solid State Circuits, vol. 21, pp. 73–85, Feb. 1986.

4. K. K. Parhi, "A systematic approach for design of digit-serial processing
tectures," *IEEE Trans. on Circuits and Systems*, vol. 38, no. 4, pp. 358
April 1991.

5. R. I. Hartley and K. K. Parhi, *Digit-Serial Computation*. Kluwer, 1995.

6. N. E. Weste and K. Eshraghian, *Principles of CMOS VLSI design: A Sy
Perspective*. Addison-Wesley, 1993.

III. 驗證 9.25 投影片 Signed Binary Digit Arithmetic 架構.

並設計 — 4-digit LSD-first adders