CE 6305 Quiz1, spring 2012

Work in groups of 2, 3, or 4. Add your names at the top of this page.
Attempt any subset of questions as time will allow.

1. An error detection system is to be added to an unsigned binary adder. The idea is to take each of the operands and multiply them by 3 before the addition. An error is signaled by the system if the result of the addition is not divisible by 3.

   (a) If the operands are k bits wide, how many bits of result must be generated by the adder?

   Answer
   To test all the bits of the adder, k+2 bits will be needed.

   (b) Describe a fast way to multiply the operands by 3.

   Answer
   $y = x + x << 1$, add x to x left shifted one place.

   (c) Characterize the kinds of errors that can be detected by this system.

   Answer
   Any error that can be characterized by the addition of a constant error term that is not divisible by 3 can be detected. For example, if the answer should be $(a \times 3 + b \times 3)$ but it is $(a \times 3 + b \times 3) + const$ where const is an integer not divisible by 3, then the error will be detected.

   But if two bits are in error, such errors may not be detected.

   For example, if the answer should be $156_{10} = 10011100_2$ and it is $10011111_2 = 159_{10}$ the result is divisible by 3: $159 \div 3 = 53$

   (d) Can all single bit errors be detected? Explain.

   Answer
   Yes, 1-bit errors can be detected by this system. For example, if the correct answer is x and any one bit is in error, then we would get the answer $x + 2^n$ If x divides by 3 then the erroneous result will not divide by three.

   (e) Can this error detection method be generalized to detect n bit errors? Explain.

   Answer:
   Any unidirectional error with arithmetic weight not exceeding $a - 1$ can be detected by a low-cost product code that uses a check modulus of $A = 2^a - 1$.

   A low-cost product code is one where both operands in an addition are pre-multiplied by a constant of the form $A = 2^a - 1$. These are low-cost since we can get $A \times x$ by shifting x to the left a places and subtracting x from the result.

   For example, $A = 15$ can detect additive errors of weight 2 or 3. Additive errors of form 2+4 = 12, 128 + 4 = 132, 16+4+2 = 22, 256 + 16 + 2 = 274 are all detectable because the added error terms (12, 132, 22, 274) are not divisible by 15.

2. An integer multiplier takes two k digit unsigned numbers X, and Y, in radix r, and produced a $2k$ digit result, Z.

   (a) If a shift of one digit position takes T_s time units, and an add of two m digit numbers takes mT_a time units, form an equation for the time taken by a shift-add multiplier.
   Answer
   There will be k cycles in which a k bit add and a one bit shift takes place, so the total time is $k^2T_a + k \times T_s$.

   (b) Now consider a multiplier that works as follows: Each of the input numbers is partitioned into two halves:

$$
\begin{aligned}
X &= Ar^{k/2} + B \\
Y &= Cr^{k/2} + D \\
& \quad \text{Intermediate results are computed as follows:} \\
U &= (A+B) \times (C+D) \\
V &= A \times C \\
W &= B \times D \\
& \quad \text{Then the result is formed as follows:} \\
Z &= Vr^k + (U - V - W)r^{k/2} + W
\end{aligned}
$$

   Show the technique in operation for $r = 2$, $k = 16$,
   $X = 0110\ 1001\ 0011\ 1100_2 = 693C_{16}$,
   $Y = 0011\ 0001\ 1111\ 1010_2 = 31FA_{16}$
   Answer
   $693C \times 31FA = 148B4098_{16} = 344670360_{10}$
   $A = 69_{16}$, $B = 3C_{16}$, $C = 31_{16}$, $D = FA_{16}$,
   $U = (69 + 3C) \times (31 + FA)$ all base 16 $= (A5) \times (12B) = C0B7_{16}$
   $V = 69 \times 31 = 1419_{16}$
   $W = 3C \times FA = 3A98_{16}$
   $Z = 1419 * 2^{16} + (C0B7 - 1419 - 3A98) * 2^8 + 3A98 = 14190000 + 720600 + 3A98 = 148B4098_{16}$
   Again using decimals:
   $A = 105_{10}, B = 60_{10}, C = 49_{10}, D = 250_{10}$
   $U = (105 + 60) \times (49 + 250) = 165 \times 299 = 49335_{10}$
   $V = 105 \times 49 = 5145_{10}$
   $W = 60 \times 250 = 15000_{10}$
   $Z = 5145 \times 2^16 + (49335 - 5145 - 15000) \times 2^8 + 15000 = 337182720 + 7472640 + 15000 = 344670360_{10}$

(c) Assuming k is even, form an equation for the time to complete the multiply using this technique.

Answer

The total is the time for 3 multiplies and 6 adds.

(d) Now assume that $k = 2^p$ for integer p, so that we can apply the technique in (b) above recursively, and develop an equation for the time complexity of the resulting multiplier.

Answer

If the time to multiply two k bit numbers is $T(k)$ and the time to add two k bit numbers is cK, $(c = Ta)$, then using the above technique gives a time of:

$$\begin{aligned} T(k) &= 3T(k/2) + ck \\ &= O(k^{1.58}) \end{aligned}$$

3. Consider a k bit by k bit multiplier based on the idea of Wallace Trees in which two or three bits are combined in a half-adder or a full-adder as early as possible until there are two bits remaining in the fewest number of leftmost columns.

   Form an equation in terms of k for the total number of bits that must be added by the Wallace Tree in the middle column of the multiplier.

   This equals the number of bits in the tallest column of the input dot diagram (before the first stage of full adders) plus the numbers of carry bits into that column.

   For example, in a 5x5 multiplier there are 5 bits in the middle column of the input dot diagram and there are c5 carries into the middle column within the Wallace Tree, making a total of 5+c5 bits to be added in that column. The Wallace Tree for that column must add 5+c5 bits, producing 2 bits that are added in the final CLA. You must find an equation for the total number of bits that must be added within the tallest (middle) column of the multiplier.

   Answer

4. Estimate the numbers of half adders needed (a full adder counts as two half adders) in an array multiplier, a Wallace Tree multiplier, and a tree multiplier using CSA 4:2 units for a 32 but multiplier.

   Answer

5. In a 16 bit array multiplier Booth Radix 4 recoding is to be used so that the number of partial products is halved. Explain how the recoded multiplier value is integrated into the array multiplier design.

Answer

Each partial product added in the array must be selected from 0, a, $-a$, $2a$, and $-2a$. These values could be pre computed and stored in registers. Then $k-$bit muxes would be used in each row of the array ($k/2$ rows) to select the partial products.