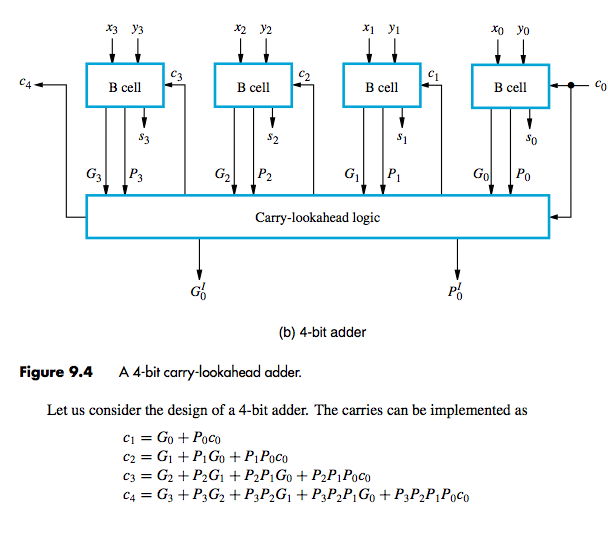
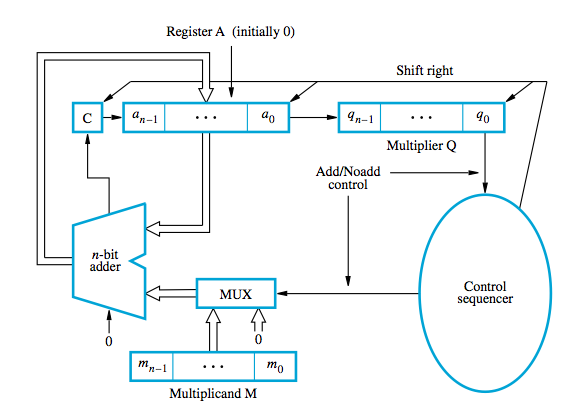
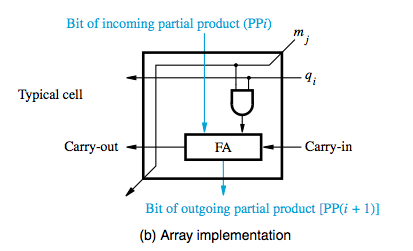
* Overflow is detected when signs of two operands are the same but the output sign is different
* Overflow = *xn*−1*yn*−1*sn*−1 + *xn*−1*yn*−1*sn*−1
* It can also be shown that overflow occurs when the carry bits *cn* and *cn*−1 are different. (See Problem 9.5.) Therefore, a simpler circuit for detecting overflow can be obtained by implementing the expression *cn* ⊕ *cn*−1 with an XOR gate.
* Understand Ripple Adder and Carry Look Ahead adder
* 
* G = xy and P = (x+y)
* The product of two, unsigned, *n*-digit numbers can be accommodated in 2*n* digits, so the product of the two 4-bit numbers in this example is accommodated in 8 bits, as shown.
* SEE FIG 9.6
* What is a partial product? P.344
* Array Multiplier, sequential circuit multiplier
* Know this Array multiplier below
* 
* Know how this sequential circuit multiplier works
* Go over multiplication of signed numbers p.348
* How do you do a sign extension to partial products in hardware? p.348 problem 6.19
* Booth Algorithm
* In general, in the Booth algorithm, −1 times the shifted multiplicand is selected when moving from 0 to 1, and +1 times the shifted multiplicand is selected when moving from1 to 0, as the multiplier is scanned from right to left
* See worst case booth multiplier alternating 1s and 0s p. 351
* Benefits of booth: handles (+/-) numbers uniformly and is more efficient
* Bit pair recoding only requires n/2 summands
* Bit pair recoding is derived from Booths
* RIPPLE CARRY ARRAY VS. CARRY SAVE ARRAY. Does the same thing but less delay
* Delay saved is in problem 6.22
* 3-2 reducer and 4-2 reducer don’t need to know?
* Da
* Floating Point Operations: Assumption is A (+/-/\*//) B

Given two Floating points A and B.

Where SA = sign of A, EA = exponent of B, MA = mantissa of A, SB = sign of B, EB = exponent of B, MB = mantissa of B, SR = sign of result, MR = mantissa of result, ER = exponent of result

Multiplication of Floating points

Multiply SA and SB to get SR which is just an XOR gate. SA = 1 means negative

Add EA and EB to get ER

Multiply MA and MB to get MR.

Normalize MR: Determine

Division of Floating points

Do and XOR of SA and SB to get SR. SA = 1 means negative

EA – EB = ER

Divide MA by MB to get MR

Normalize MR: Deter