Arithmetic Circuits

High-Speed Multipliers

Dae Hyun Kim

EECS
Washington State University

Multiplication

- We will use the unsigned binary number system.
  - (Signed multiplication is similar to unsigned multiplication.)

- Example

```
          10101111
        × 11010111
        ----------
          10101111
         10101111
        10101111
       00000000
      10101111
     00000000
    10101111
   10101111
----------------
1001001011111001
```

Theory

- Suppose we multiply two n-bit unsigned binary numbers, A and B.
- How many bits do we need to represent the result?
  - A, B: $[0, 2^n - 1]$
  - $M = A * B$: $[0, 2^{2n} - 2^{n+1} + 1]$
    - The max. value that can be represented by $2n - 1$ bits: $2^{2n-1} - 1$
    - $(2^{2n} - 2^{n+1} + 1) - (2^{2n-1} - 1) = 2^{2n-1} - 2^{n+1} + 2 > 0$
    - The max. value that can be represented by $2n$ bits: $2^{2n} - 1$
    - $(2^{2n} - 2^{n+1} + 1) - (2^{2n} - 1) = -2^{n+1} + 2 < 0$
  - Therefore, we need $2n$ bits to represent $A * B$.

What to Calculate to Obtain $A * B$

- Add the partial products.

```
           10101111     A
         × 11010111     B
        ------------
           10101111  ---->  Partial product (A * B_0)
          10101111   ---->  Partial product (A * B_1)
         10101111
        00000000
       10101111
      00000000
     10101111
    10101111  ---->  Partial product (A * B_{n-1})
    -----------------
    1001001011111001
```

- How many partial products do we add?
  - n

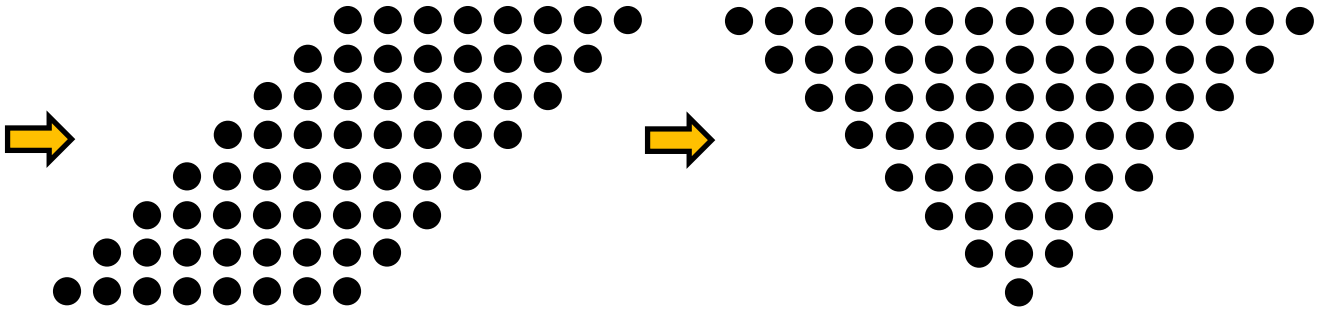
Sequential Addition

- Add $A * B_0$ and $A * B_1$
  - $PS_{1:0} = A * B_0 + A * B_1$
- Then, add $PS_{1:0}$ and $A * B_2$
  - $PS_{2:0} = PS_{1:0} + A * B_2$
- ...
- Add $PS_{n-2:0}$ and $A * B_{n-1}$: $PS_{n-1:0}$
  - How many adders do we need: 1
  - Delay: $(n-1) \cdot d_A$ where d_A is the delay of an $(n+1)$-bit adder.

Parallel Addition

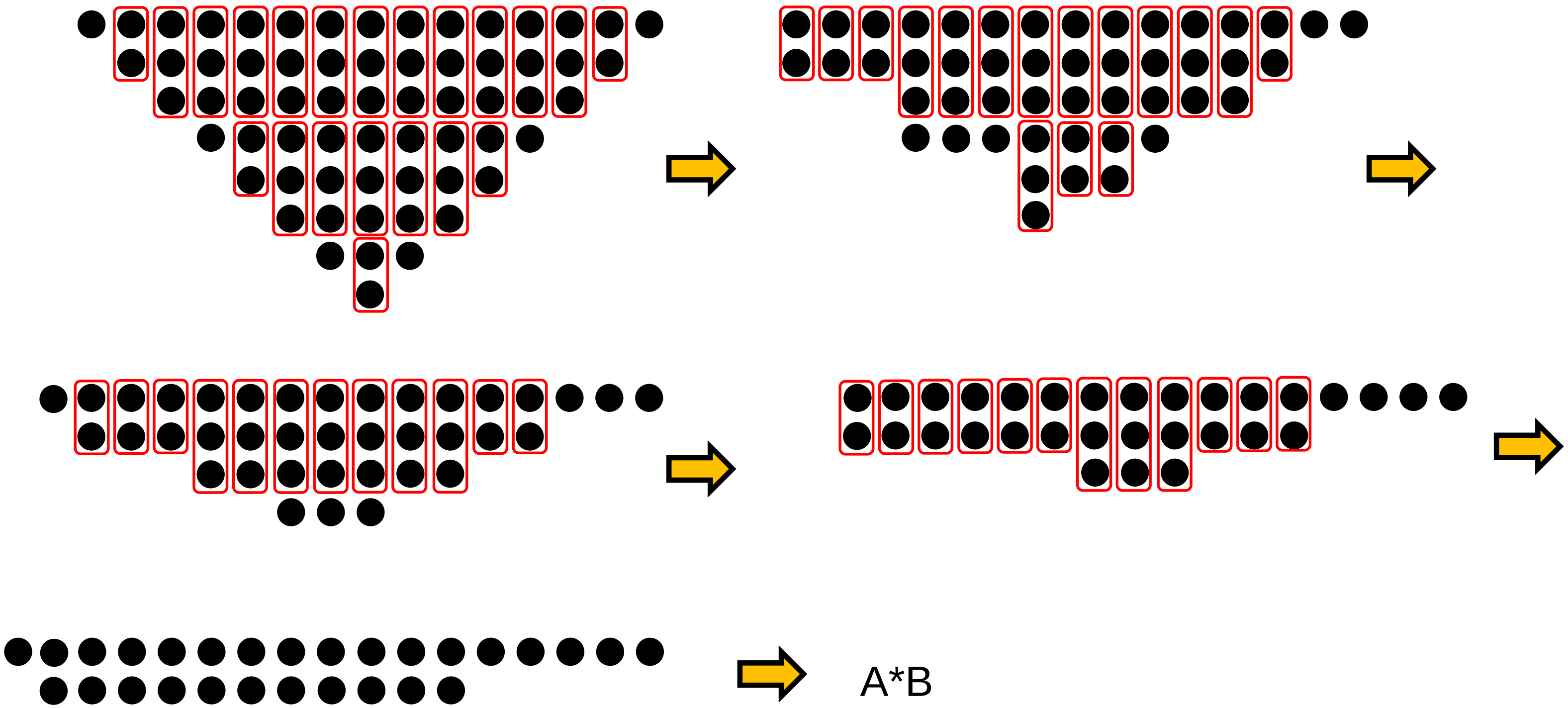
- Add all the bits

```
          10101111
        × 11010111
        ----------
          10101111
         10101111
        10101111
       00000000
      10101111
     00000000
    10101111
   10101111
----------------
1001001011111001
```



Parallel Addition

- Use HAs and FAs



Carry-propagation adder (CPA)

Modified Booth Encoding

- How to reduce # partial products (for A*X)

$$\ldots\ x_7\ x_6\ x_5\ x_4\ x_3\ x_2\ x_1\ x_0\ (x_{-1})$$

$$y_7 y_6 \quad y_5 y_4 \quad y_3 y_2 \quad y_1 y_0$$

x_i	x_{i-1}	x_{i-2}	y_i	y_{i-1}	Operation	Comments
0	0	0	0	0	+0	string of zeros
0	1	0	0	1	+A	a single 1
1	0	0	$\bar{1}$	0	-2A	beginning of 1's
1	1	0	0	$\bar{1}$	-A	beginning of 1's
0	0	1	0	1	+A	end of 1's
0	1	1	1	0	+2A	end of 1's
1	0	1	0	$\bar{1}$	-A	a single 0
1	1	1	0	0	+0	string of 1's

Modified Booth Encoding

- Example

```
   A        0 1 1 0 1 1 0 1
  *X        0 0 1 0 1 0 1 1 (0)
  ----------------------------
   A        0 1 1 0 1 1 0 1
  *Y        0 1 0 1’0 1’0 1’
  ----------------------------
                           -A
                       -A
                   -A
               +A
  ----------------------------
            -0 1 1 0 1 1 0 1
        -0 1 1 0 1 1 0 1
    -0 1 1 0 1 1 0 1
+0 1 1 0 1 1 0 1
  ----------------------------
```

x_i	x_{i-1}	x_{i-2}	y_i	y_{i-1}	Operation
0	0	0	0	0	+0
0	1	0	0	1	+A
1	0	0	$\bar{1}$	0	-2A
1	1	0	0	$\bar{1}$	-A
0	0	1	0	1	+A
0	1	1	1	0	+2A
1	0	1	0	$\bar{1}$	-A
1	1	1	0	0	+0