Cairo University
Electronics and Communications Engineering

Computer Arithmetic, Lecture 5: Time bounds

Hossam A. H. Fahmy

Evaluating the time delay of a design

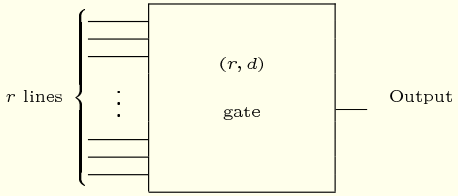
1. Modeling at the logic level. Useful for rough comparisons.
2. Transistor level simulation (sizes of transistors and buffers for the loaded gates). Does not include the long wire delays.
3. Extracted layout simulation (with wire details). Accurate area and power consumption estimation are also possible at this level.
4. Fabrication and measurement. The ultimate test for a design with a specific technology process and fabrication facilities.
5. To really show the merit of a proposed idea, simulate it over a variety of scalable physical design rule sets and fabricate one or more chips then test them.

The (r, d) Circuit Model

Winograd presents a model based on:

1. the number of digits (n) in each operand,
2. the maximum fan-in in the circuit (r), and
3. the number of truth values in the logic system (d).

The (r, d) gate evaluates any r-argument d-valued logic function in unit time.



Limitations of this first model

Winograd's (r, d) model of a logic gate is idealized in many ways:

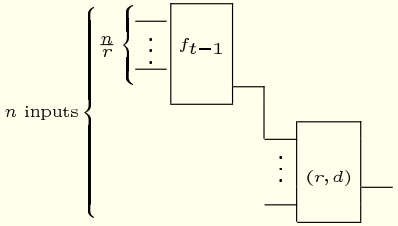
1. There is zero propagation (wire) delay between logic blocks.
2. The output of any logic block may go to any number of other logic blocks without affecting the delay (fan-out independent).
3. *Any* logical decision takes a unit delay.
4. It neglects any other mechanical or electrical considerations.

Spira's lemma

A d-valued output depending on n inputs has a time delay:

$$t \geq \lceil \log_r n \rceil$$

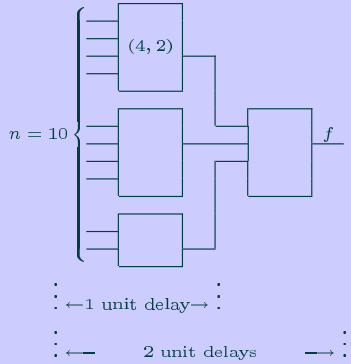
in units of (r, d) gate delays.



A simple example

Example 1 For the case of $n = 10$, $r = 4$, and $d = 2$ we get

$$\lceil \log_r n \rceil = \lceil \log_4 10 \rceil = \lceil 1.65 \rceil = 2.$$



The maximum modulus in RNS

- For an RNS system up to N numbers, $\alpha(N)$ is the number of distinct values that the largest modulus represents.
- $\log_d \alpha(N)$ is the number of d-valued lines required to represent a number for this modulus.
- An addition circuit for this modulus has $2\lceil \log_d \alpha(N) \rceil$ input lines and it needs

$$t \geq \lceil \log_r \left(2\lceil \log_d \alpha(N) \rceil\right) \rceil ,$$

What is $\alpha(N)$?

In modular arithmetic, we operate with single arguments $\mathbf{mod}_{(\mathbf{A^n})}$.

- If A is prime, then $\alpha(N)$ is simply A^n.
- If A is composite then $A = A_1 A_2 \cdots A_m$ and $\alpha(N)$ is A_i^n, where A_i is the largest element composing A.

For example,

$\alpha(10^n) = 5^n$;

for a RNS using the set $\{2^5, 2^5 - 1, 2^4 - 1, 2^3 - 1\}$,

$$\alpha(> 2^{16}) = 2^5.$$

Optimal moduli selection

Example 2 Suppose we wish to design a residue system that has $M \geq 2^{47}$.

- If we select the product of the primes, we get:
  $2\times3\times5\times7\times11\times13\times17\times19\times23\times29\times31\times37\times41 > 2^{47}$
  The $\alpha(> 2^{47})$ for this selection is 41.
- We can improve the α function by using powers of the lower order primes.
  $2^5\times3^3\times5^2\times7\times11\times13\times17\times19\times23\times29\times31 > 2^{47}$
  Here, $\alpha(> 2^{47})$ is $2^5 = 32$.

What about multiplication?

Spira's bound is applicable. Let us change the representation to minimize the number of inputs needed.

- Represent the numbers as products of prime factors or powers of prime factors.
- Add the corresponding prime factor exponents in the two numbers you want to multiply. (Subtract to divide!)
- The *Logarithmic Number System* does just that, if $a \times b = c$, then $\log a + \log b = \log c$.

The LNS

- A number X is represented by a sign bit (S_X) and $L_X = \log X$.
- For $X < 1$, add a bias to $\log X$.
- Now, $L_{XY} = L_X + L_Y$ and $L_{X/Y} = L_X - L_Y$.
- Addition and subtraction are harder, $X + Y = X(1 + Y/X)$.

It is interesting only in special applications.

Winograd and multiplication

For multiplication, we define $\beta(N)$ (akin to the $\alpha(N)$ of addition) and get:

$$t \geq \lceil \log_r (2\lceil \log_d \beta(N) \rceil) \rceil$$

Three cases are recognized:

1. Binary radix: $N = 2^n$ with $n \geq 3 \Rightarrow \beta(2^n) = 2^{n-2}$, $\beta(4) = 2$, and $\beta(2) = 1$.
2. Prime radix: $N = p^n \Rightarrow \beta(p^n) = \mathbf{max}\left(p^{n-1}, \alpha(p-1)\right)$
3. Composite powers of primes $\beta(N) = \mathbf{max}\left(\beta(p_i^{n_i})\right)$.

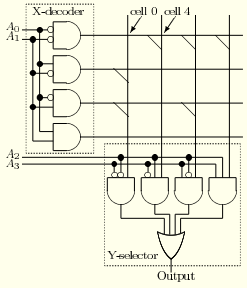
We find $\beta(N) < \alpha(N) \Rightarrow$ *the multiplication is faster than addition!*

Back to reality

- By optimizing the representation for fast addition or multiplication, a variety of other operations become much slower.
- The binary system is "complete" and comes very close to those theoretical bounds.
- Even with partial use of redundant representations, the binary system is very fast. Example: parallel multipliers use carry-save representations for multi-operand addition.

Can we just memorize the results?

The size of the table grows exponentially fast with the operand size. Hence, the table look-up is only feasible for small operands and has the following delays:



$$\begin{aligned} X\text{-decoder} &= \left\lceil \log_r \left(\frac{L}{2} \right) \right\rceil \\ \text{Memory cell} &= 1 \\ Y\text{-selector} &= \left\lceil \log_r \left(\frac{L}{2} + 1 \right) \right\rceil + \left\lceil \log_r 2^{\frac{L}{2}} \right\rceil \end{aligned}$$

We can do better

- With an overlap between the X and Y sections, the time delay is:

$$\text{ROM delay} = 2 + \left\lceil \log_r \frac{L}{2} \right\rceil + \left\lceil \log_r 2^{\frac{L}{2}} \right\rceil .$$

- When the ROM is used as a binary operator on n-bit numbers, then $L = 2 \times n$ and

$$\text{ROM delay} = 2 + \lceil \log_r n \rceil + \lceil \log_r 2^n \rceil .$$

The current modeling ignores the regularity of the memory and its limited fan-out requirements. Those features are important and favor the use of memories in some VLSI implementations.

So, which one to use?

- Except for small operand sizes, a special logic circuit is better than a table.
- Starting tables are used in division, square root, and other functions.
- More sophisticated table designs (with more than two "dimensions") yield lower time delays but they become complicated.

Multiplexers

- A single m-to-1 multiplexer is considered to take only one *FO4* delay from its inputs to the output assuming it is realized using CMOS pass gates. This assumption for the multiplexer is valid up to a loading limit.
- Small m is the usual case in VLSI design since multiplexers rarely exceed say a 5-to-1 multiplexer.
- For a single multiplexer the delay from the select lines to the output is bounded by 2 *FO4* delays.

n-bit multiplexers

- A series of m to 1 multiplexers connected to form a larger n-bit multiplexer heavily loads its select lines.
- About each four multiplexers should have a buffer and form a group together.
- Four such groups need yet another buffer and form a super group and so on.

The delay of the selection is then $\lceil \log_4(n) \rceil + 1$.

Shifters

- Combinational shifters are either done by a successive use of multiplexers or as a barrel shifter realized in CMOS pass transistors.
- The delay of an n-way shifter from its inputs to its outputs is $\lceil \log_2(n) \rceil$ *FO4* delays.
- The select lines are heavily loaded as in the case of multiplexers but their delay is smaller than the delay from the inputs to the outputs in the shifter.

Modeling summary

Part	Delay
Multiplexer, input to output	1
Multiplexer, select to output	$\lceil \log_4(n) \rceil + 1$
Shifter	$\lceil \log_2(n) \rceil$
Memory	$2 + \lceil \log_r \frac{n}{2} \rceil + \lceil \log_r 2^{\frac{n}{2}} \rceil$
Spira's bound (no design details)	$\lceil \log_r(n) \rceil$