Greatest Common Factor Calculator

Group Members
Scott Noel, Carlos Marchani and How-Ying Liou
Project Description

- Two 7-bit unsigned inputs
- One 7-bit unsigned GCF output
- Calculates prime divisors between 0 and 127
- Shift and subtract division
- Shift and add multiplication
### Example

**Inputs**

<table>
<thead>
<tr>
<th></th>
<th>24</th>
<th>36</th>
</tr>
</thead>
<tbody>
<tr>
<td>First divisor</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>Remainders = 0?</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Divisor</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>Remainders = 0?</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Divisor</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Remainders = 0?</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Restore dividends</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>Divisor ++</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Divisor &gt; Dividend</td>
<td>STOP</td>
<td></td>
</tr>
</tbody>
</table>

\[ \text{GCF} = 2 \times 2 \times 3 = 12. \]
Main Algorithm Diagram

Loop start

- Compare dividend A & divisor
  - A and B both > divisor?
    - Yes
      - Divide dividend A by divisor => quotient A
        - Both remainders = 0?
          - No
            - divisor = next prime divisor
          - Yes
            - dividend A = quotient A
            - dividend B = quotient B
            - product = product * divisor
        - No
          - Stop
    - No
      - Stop

- Compare dividend B & divisor
  - No
    - Stop
Prime Number Generator

2 3 5 7 11...127
\ /
\ /
\ /
1 2 2 4
Meg File for Divisor Generator

INPUTS: RESTART I1 I2 I3 I4 I5 I6 I7 COUNT;
OUTPUTS: O1 O2 O3 O4 DONE;
RESET ON RESTART TO s2(O3);
s1: IF COUNT THEN s2(O3) ELSE s1(O4);
s2: case (I1 I2 I3 I4 I5 I6 I7 COUNT)
0 0 0 0 0 1 0 1 => s1(O4);
0 0 0 0 1 0 1 1 => s2(O3);
0 0 0 0 1 1 1 1 => s4(O2);
0 0 0 1 1 0 1 1 => s4(O2);
0 0 1 0 0 1 1 1 => s4(O2);
0 1 1 1 1 1 1 1 => s6(O2 O3);
1 0 1 0 1 0 1 1 => s4(O2);
1 1 1 1 1 0 1 1 => s6(O2 O3);
1 0 0 1 0 0 1 1 => s6(O2 O3);
1 1 0 0 1 1 1 1 => s4(O2);
1 1 0 1 1 0 1 1 => s4(O2);
endcase => ANY;
Division Algorithm

Divisor

7-bit ALU

7 bits

Remainder

Shift Right
Shift Left
Write

14 bits

Control Test
Multiplication Algorithm

Multiplicand

7-bit ALU

Product

Shift Right Write

Control Test

7 bits

14 bits
One-Bit Adder Cell
Spice Analysis

Graph 1:
- Voltage vs. Time
- Graph shows a transition from 0 to 5V over time.

Graph 2:
- Voltage vs. Time
- Graph shows a decay from 5V to 0V over time.
7-bit ALU

Crystal Timing Analysis: Longest path is 41.91 ns
Timing Diagram for Division
Multiplication Subcell
Floor Plan
PLA Information

- 5 PLAs
- Main PLA – gets input from datapath PLAs, controls datapath PLAs and assorted signals
- Prime Generator PLA (6 states) - outputs divisor increment
- Division PLA (27 states) – controls division
- Multiplication PLA (20 states) – controls multiplication
- Counter (7 states) – counts to 7 and restarts, used internally
Division PLA State Diagram

- **IDLE**
  - Setup (5 states)
  - Add (3 states)
  - Subtract
  - Shift (2 states)
  - Shift (2 states)

- **Wait** (2 states)
  - Shift (2 states)
  - Add (3 states)
  - Subtract
  - Shift (2 states)

- **Shift (2 states)**
  - /Count
  - /Bit0=0
  - /Bit0=1

- **Add (3 states)**
  - Bit13/compute

- **Subtract**
  - /sub compute

- **Setup** (5 states)
  - PLA1/compute restore

- **Finish**
  - done
Remainders ≠ 0, restore dividends
Remainders = 0, write dividends into registers, multiply

Divide

Compare

Initialize division

Initialize multiplication

Divisor > min(Dividends)

Generate prime number

Finish

DONE

IDLE

Finish
Key Features

- Simultaneous division to increase throughput
- Save extraneous division cycles by early termination
- Running product
- Efficient prime divisor generator
- Master-slave shift structures