



**FPGA-Based Design and Implementation of the
Greatest Common Divisor using**

SPARTAN⁶



**Presented by: Ibrahim Hazmi
Supervised by: Dr. Mihai Sim**



**University
of Victoria**

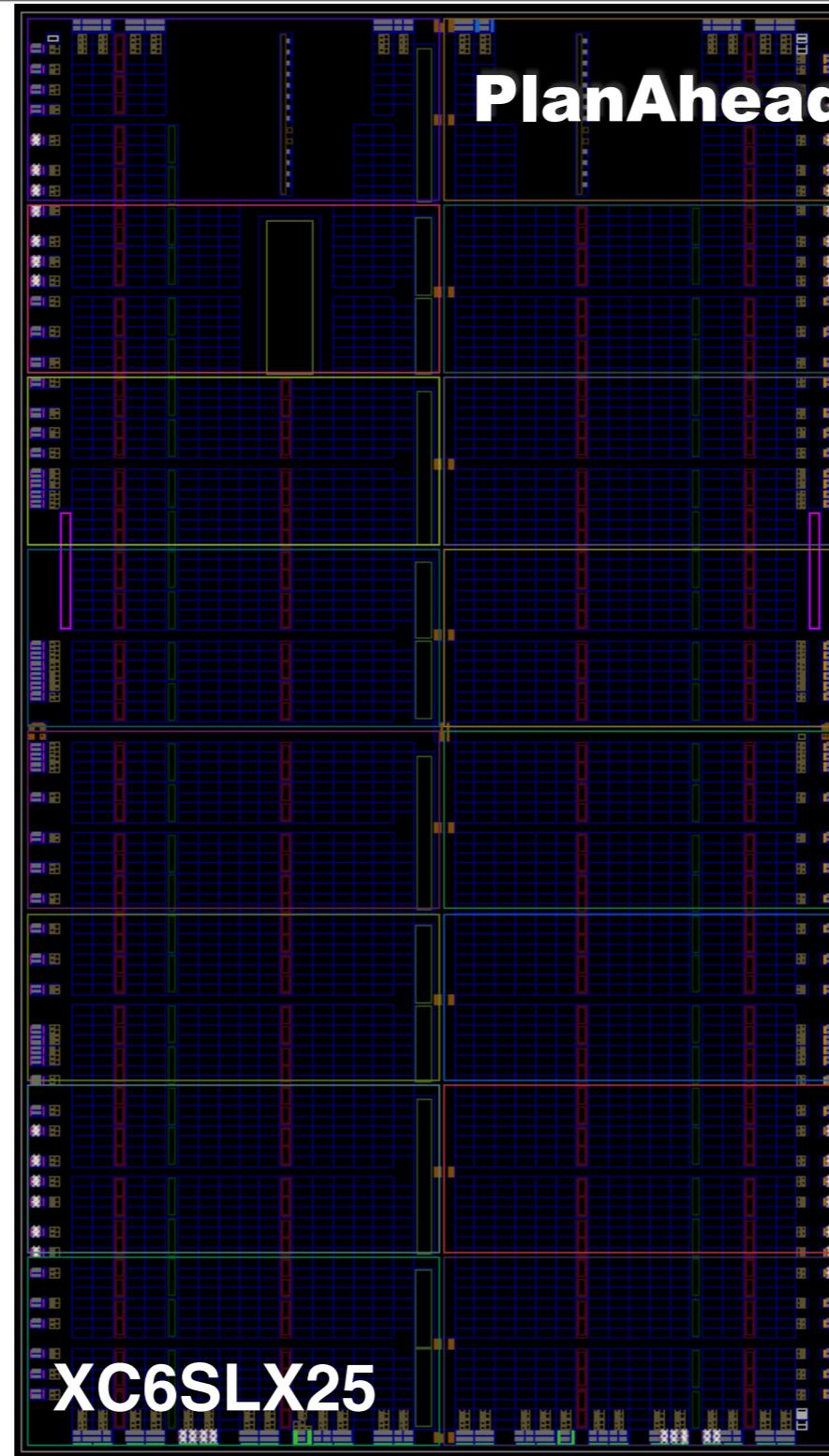
Design Problem & Motives

- Design 16-bit GCD (the Greatest Common Divisor) and Implement it on Xilinx Spartan6 FPGA.
- It is challenging is to design such circuit and examine the area-speed tradeoff in order to get the result at reasonable speed with optimal utilization of the FPGA resources.
- So, the design came in different Levels:
 - High Behavioural Level
 - ASM Behavioural Level
 - Direct Structural Level
 - Optimized Structural Level

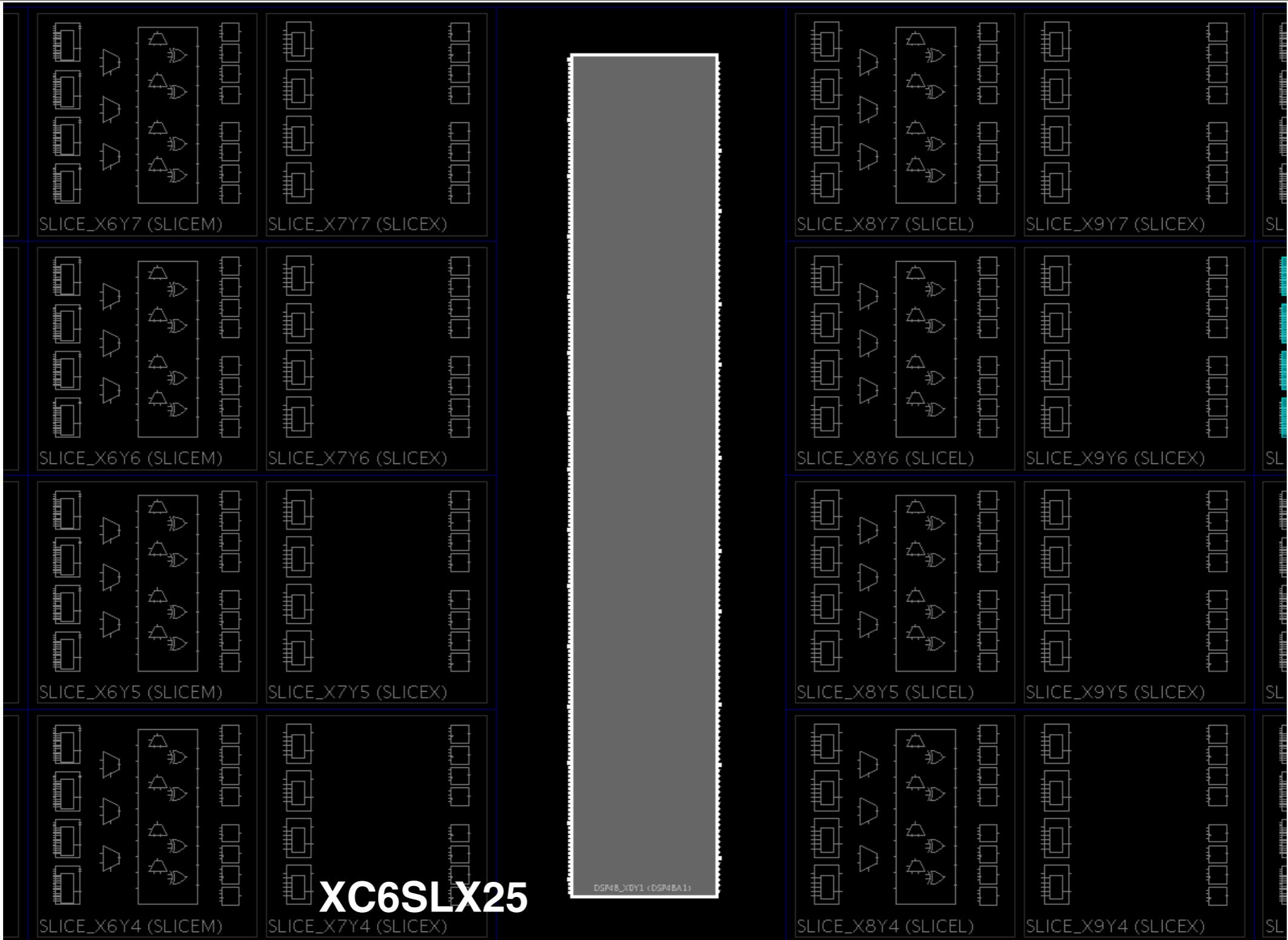
Outline

- **Overview of Xilinx Spartan6 FPGA**
- **Overview of the GCD**
 - **Euclidean Algorithm**
- **Behavioural Level Design**
 - **High Level (While/For Loop)**
 - **ASM Level (ASM → FSM)**
- **Structural Level Design**
 - **Direct Structural (Data-path + FSM)**
 - **Optimized Structural Design (SAD)**
- **Conclusion**
- **Summary & Future Work**

Xilinx Spartan6 (Overview)



Xilinx Spartan6 (Overview)



Xilinx Spartan6 (Overview)

The diagram illustrates the internal structure of the Xilinx Spartan6 device. It features a central **CLB** block surrounded by **Logic Cells** (SLICEM and SLICEX) and **DSP Slices**. The **Logic Cells** are arranged in a grid, with labels for **SLICE_X6Y7 (SLICEM)**, **SLICE_X7Y7 (SLICEX)**, **SLICE_X8Y7 (SLICEL)**, and **SLICE_X9Y7 (SLICEX)**. The **DSP Slices** are located at the bottom, with labels for **SLICE_X6Y5 (SLICEM)**, **SLICE_X7Y5 (SLICEX)**, **SLICE_X8Y5 (SLICEL)**, **SLICE_X9Y5 (SLICEX)**, **SLICE_X6Y4 (SLICEM)**, **SLICE_X7Y4 (SLICEX)**, **SLICE_X8Y4 (SLICEL)**, and **SLICE_X9Y4 (SLICEX)**. A specific **DSP48E_XDY1 (DSP48A1)** slice is highlighted in the bottom center.

Device	Logic Cells	CLB				DSP Slices
		Slices	FFs	Max RAM	LUT6	
XC6SLX25	24,051	3,758	30,046	229	15,032	38

What is GCD ?

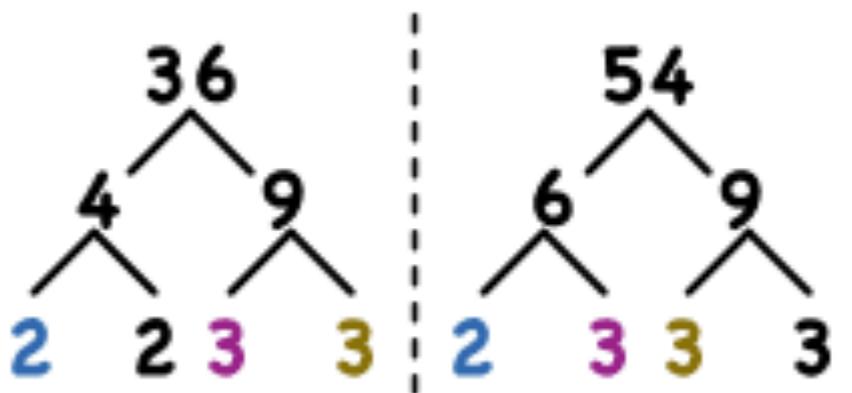
36, 54



Actually, It has many names:

- Greatest Common Divisor (GCD)
- Greatest Common Factors (GCF)
- Greatest Common Measure (GCM)
- Highest Common Divisor (HCD)
- Highest Common Factor (HCF)

Greatest Common Factor
Prime Factors



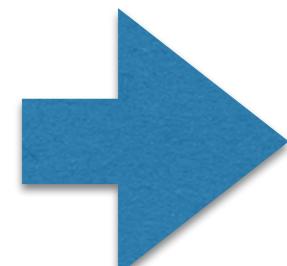
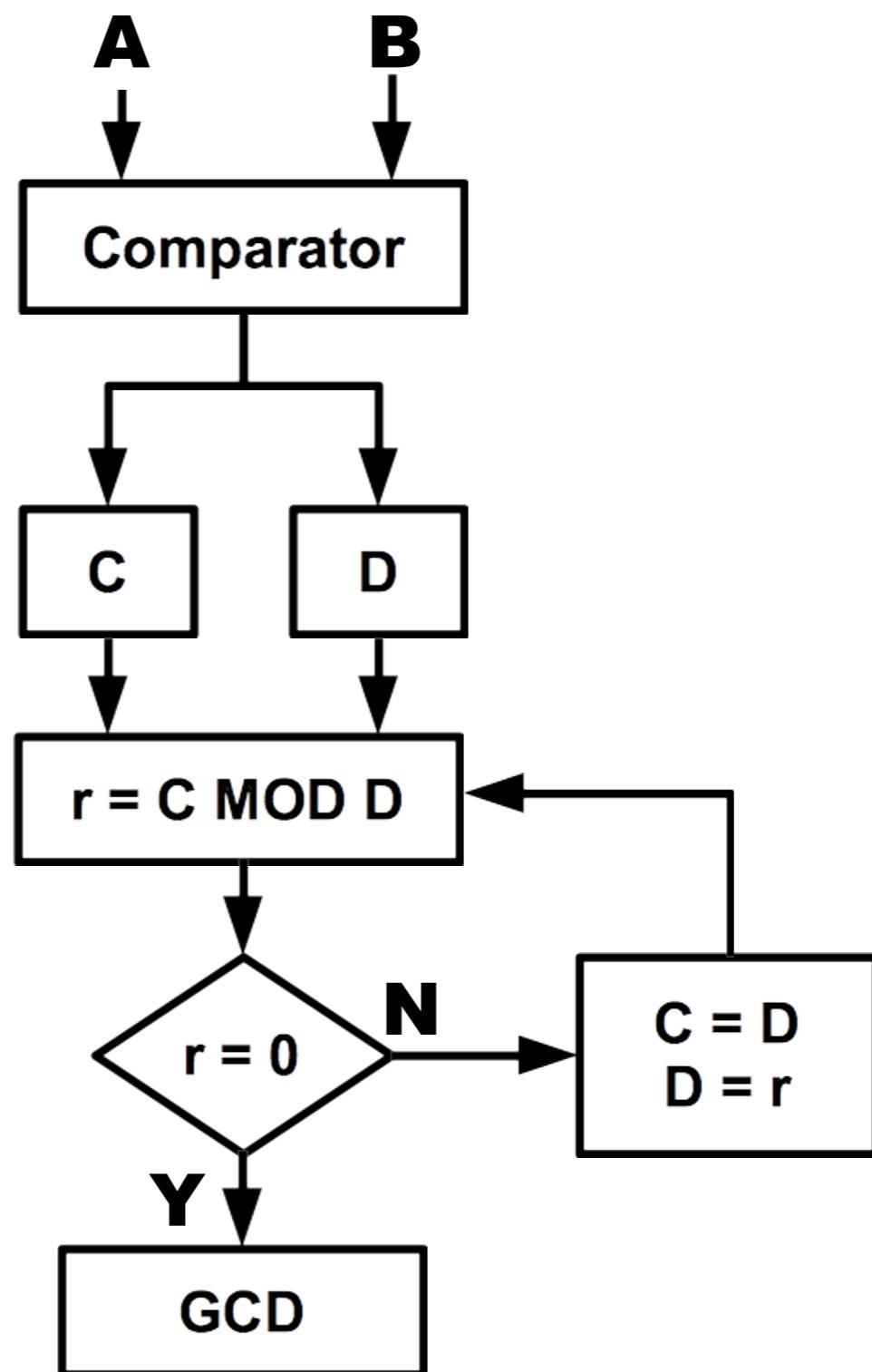
2) Shared: 2, 3, 3
3) Multiply $2 \cdot 3 \cdot 3 = 18$

Euclidean Algorithm ?

36, 54

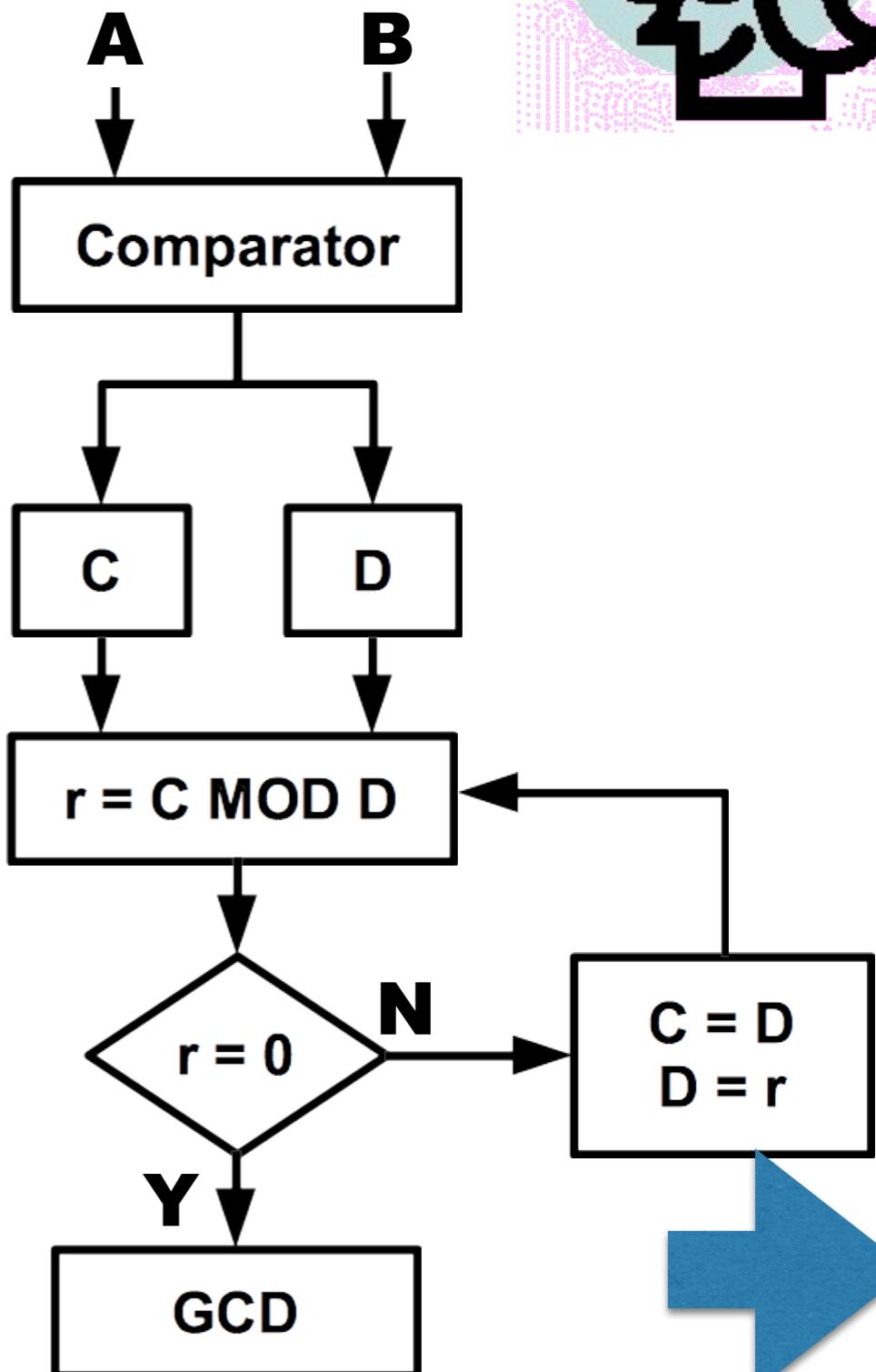
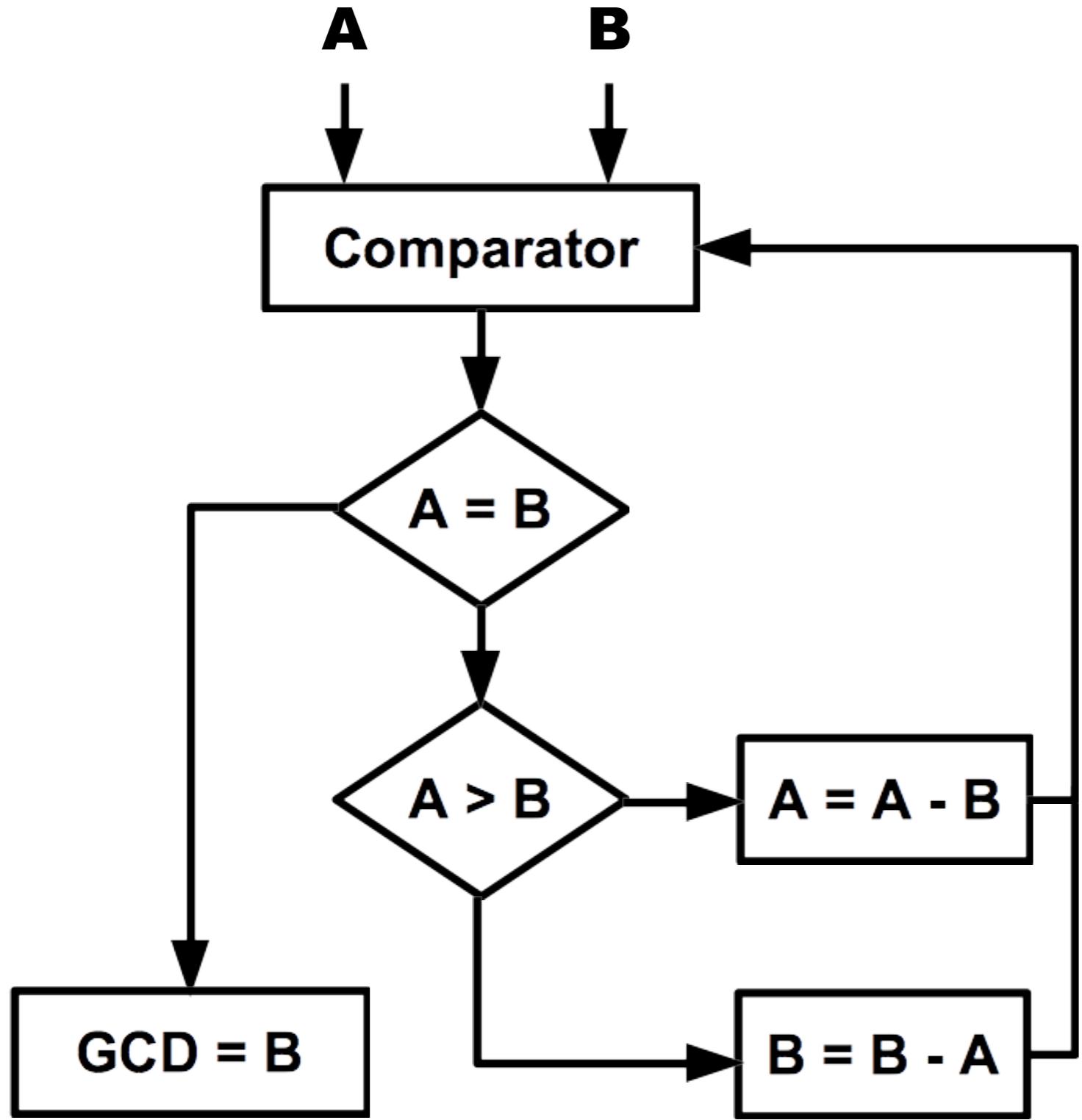


Euclidean Algorithm



Euclidean Algorithm

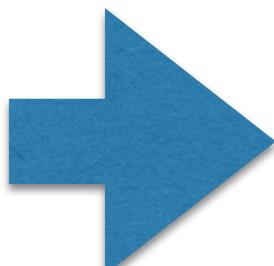
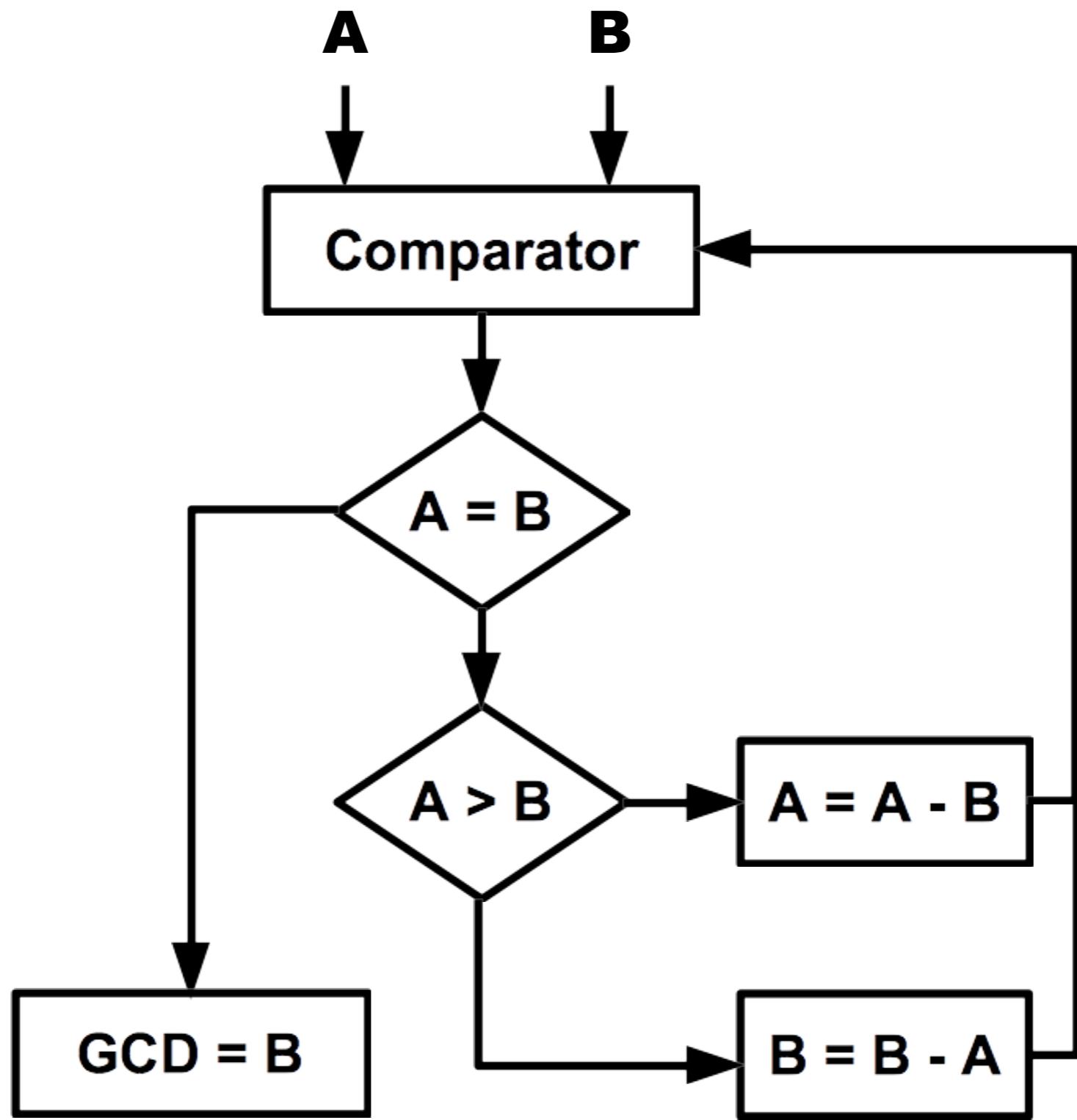
36, 54



36, 54

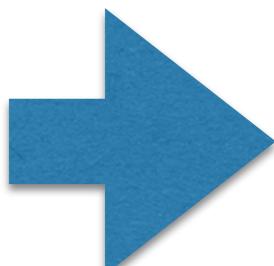
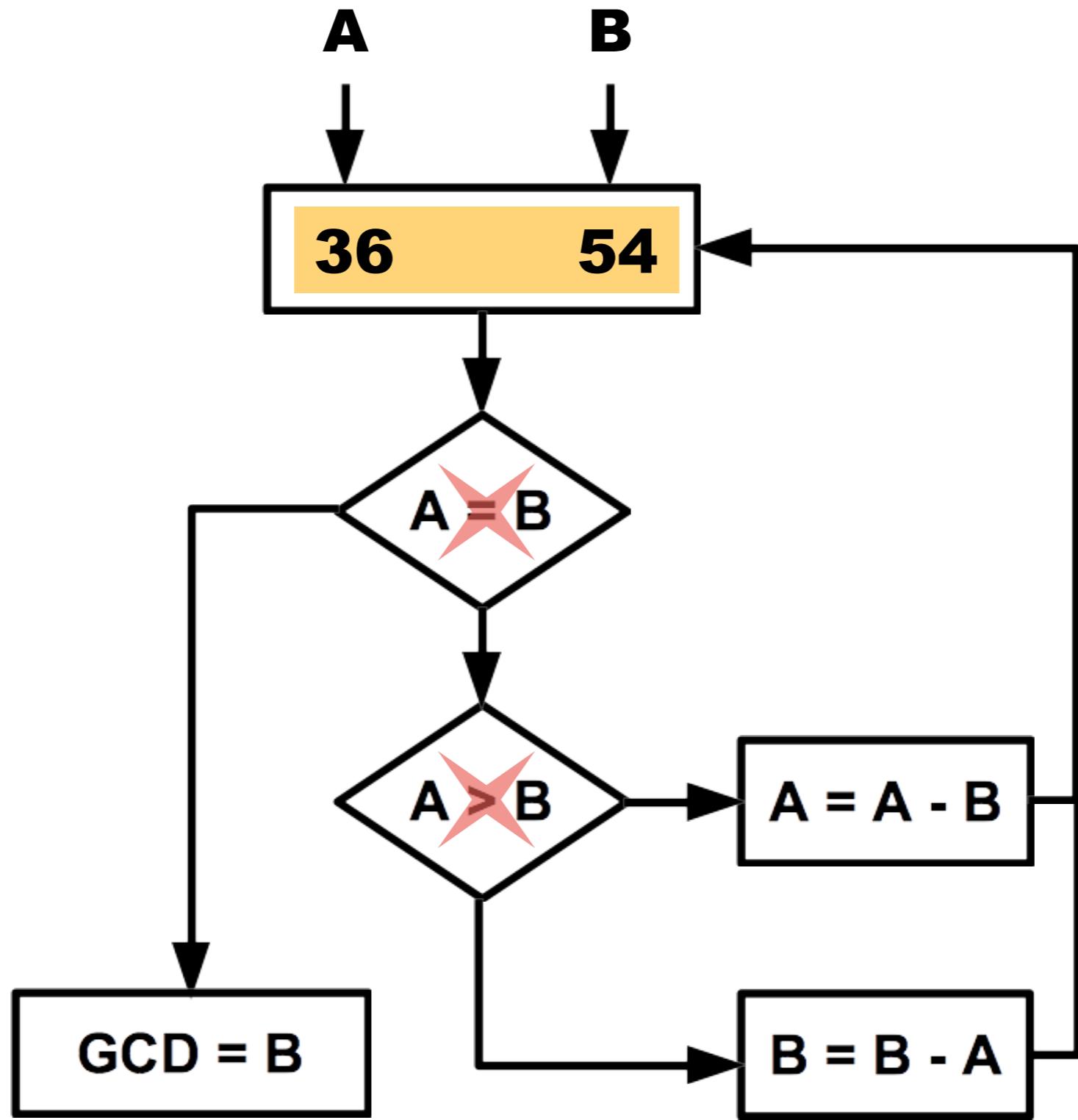


Euclidean Algorithm



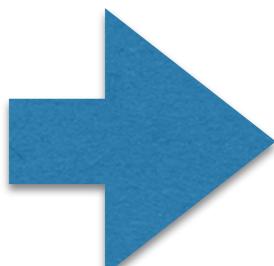
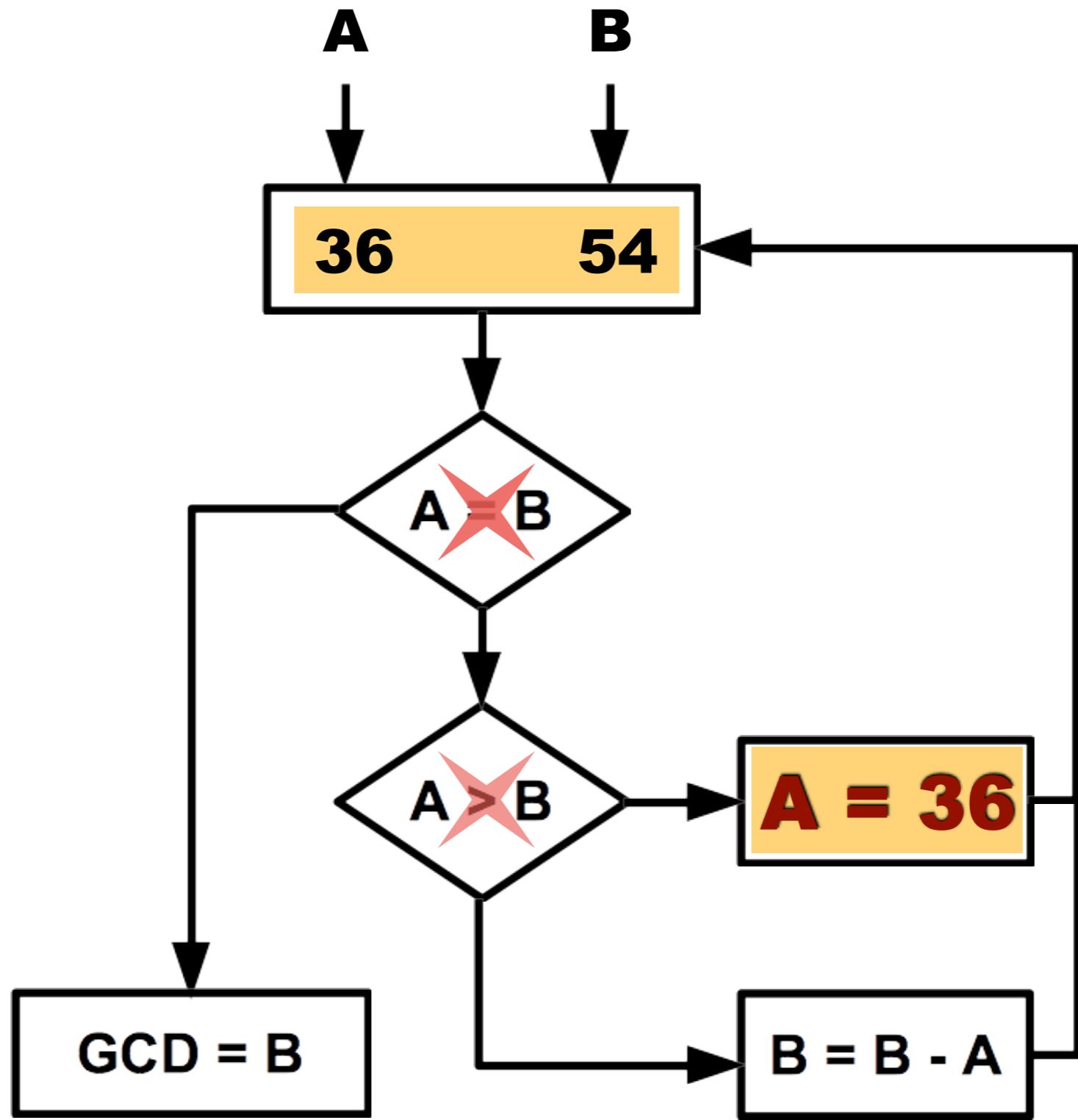
Euclidean Algorithm

36, 54



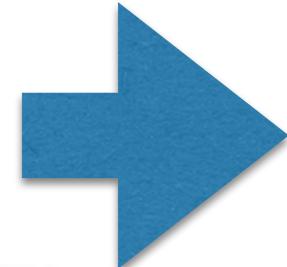
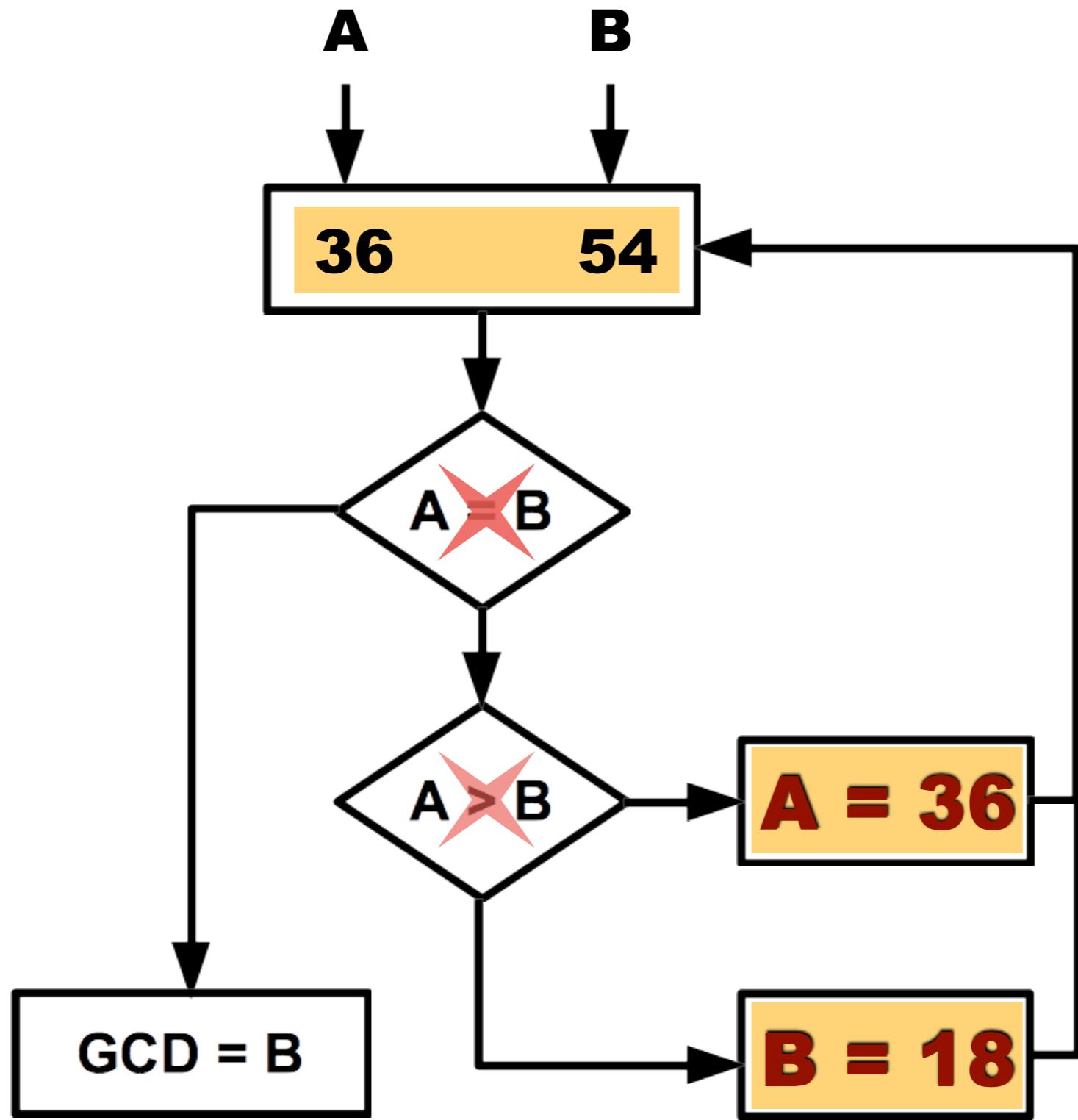
Euclidean Algorithm

36, 54



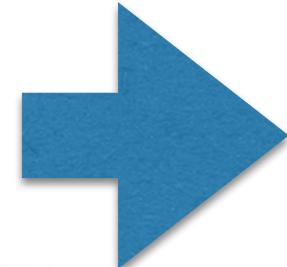
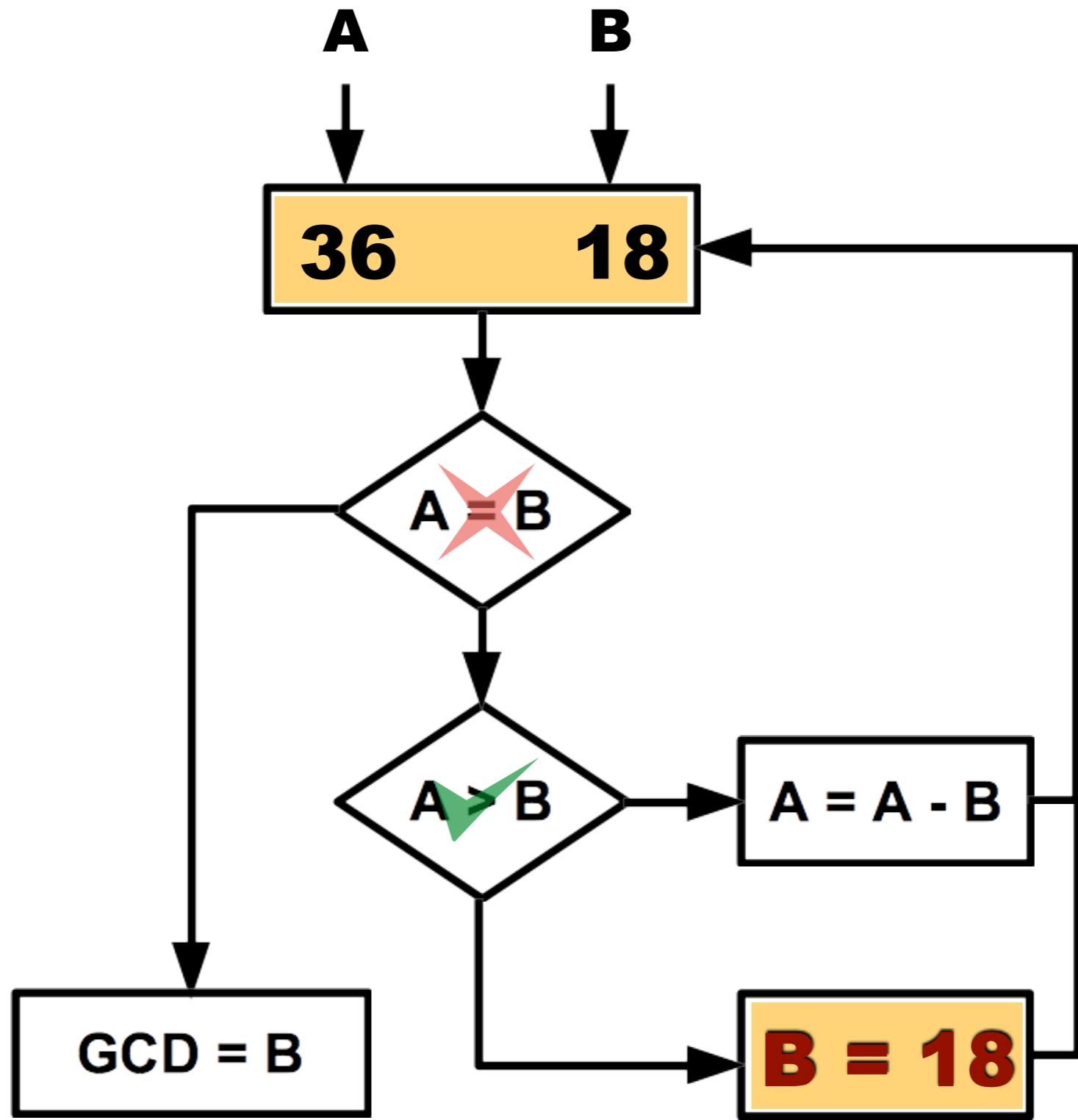
Euclidean Algorithm

36, 54



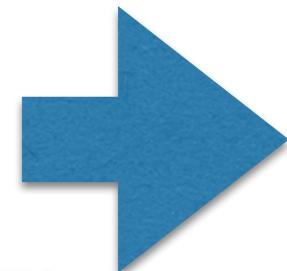
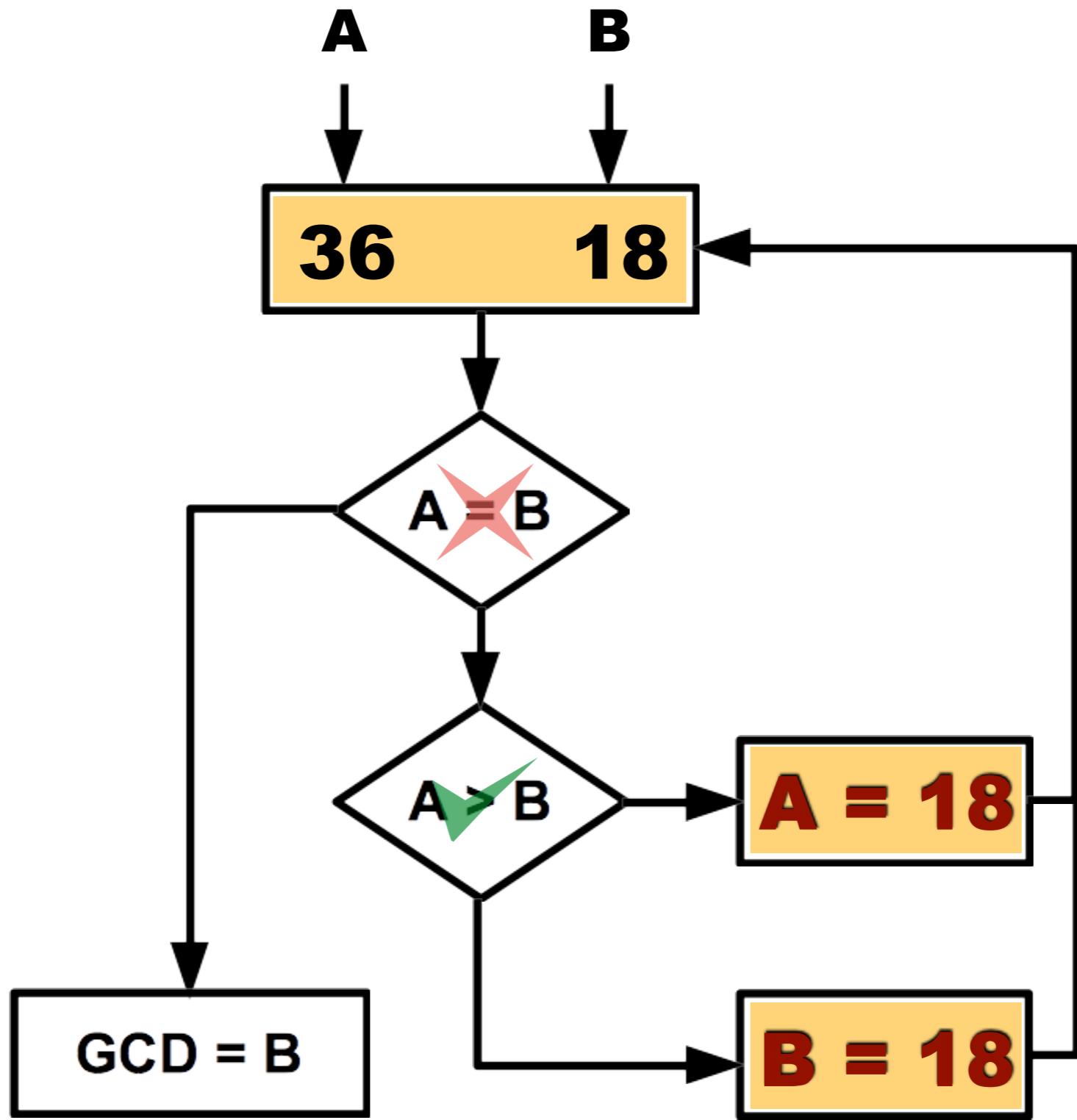
Euclidean Algorithm

36, 54



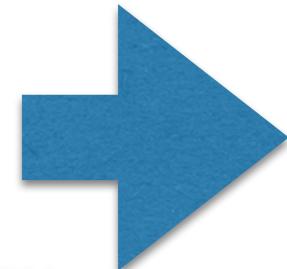
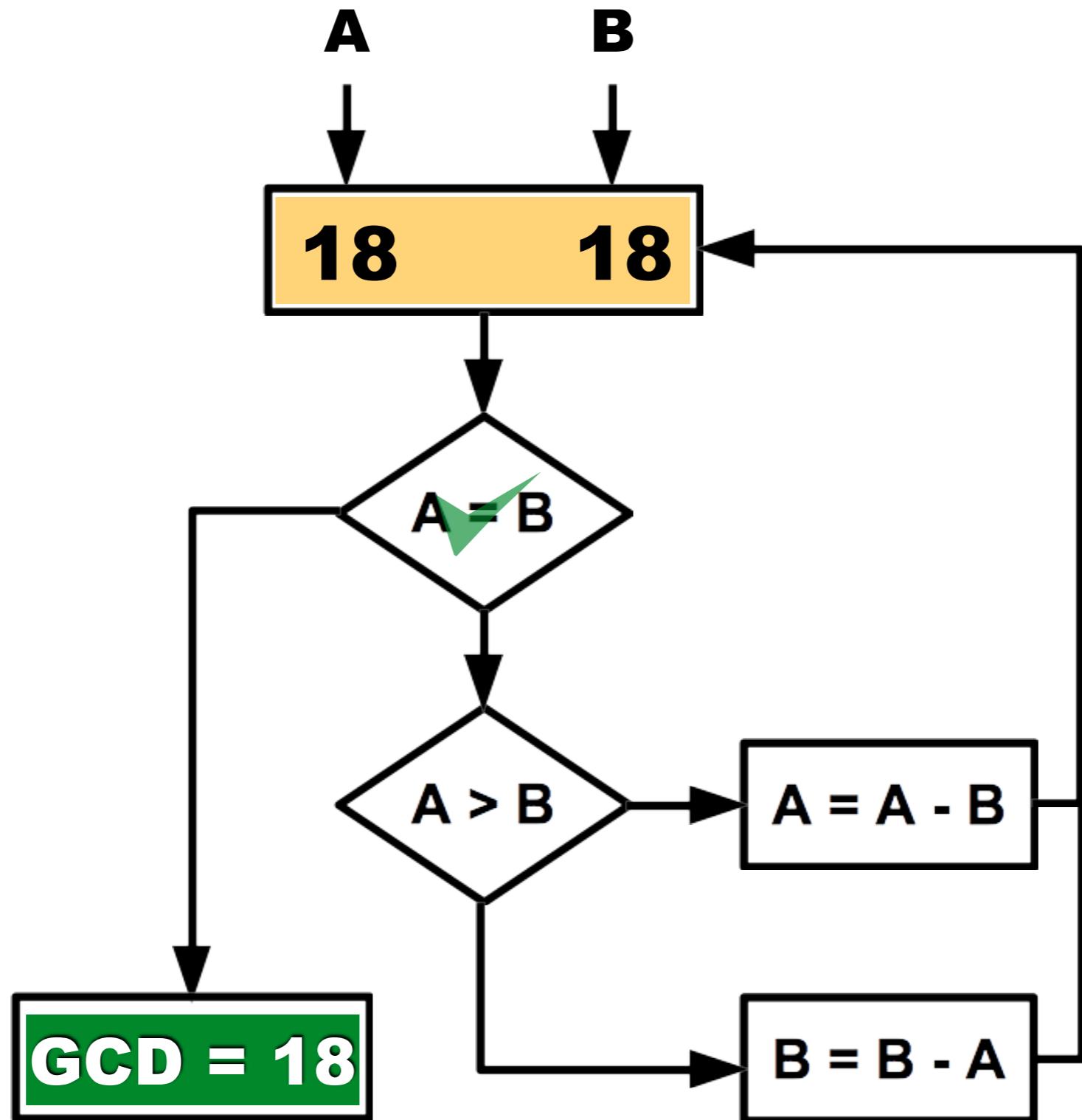
Euclidean Algorithm

36, 54



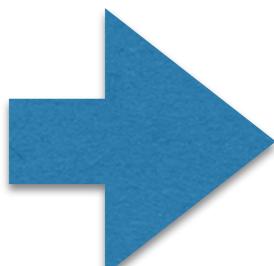
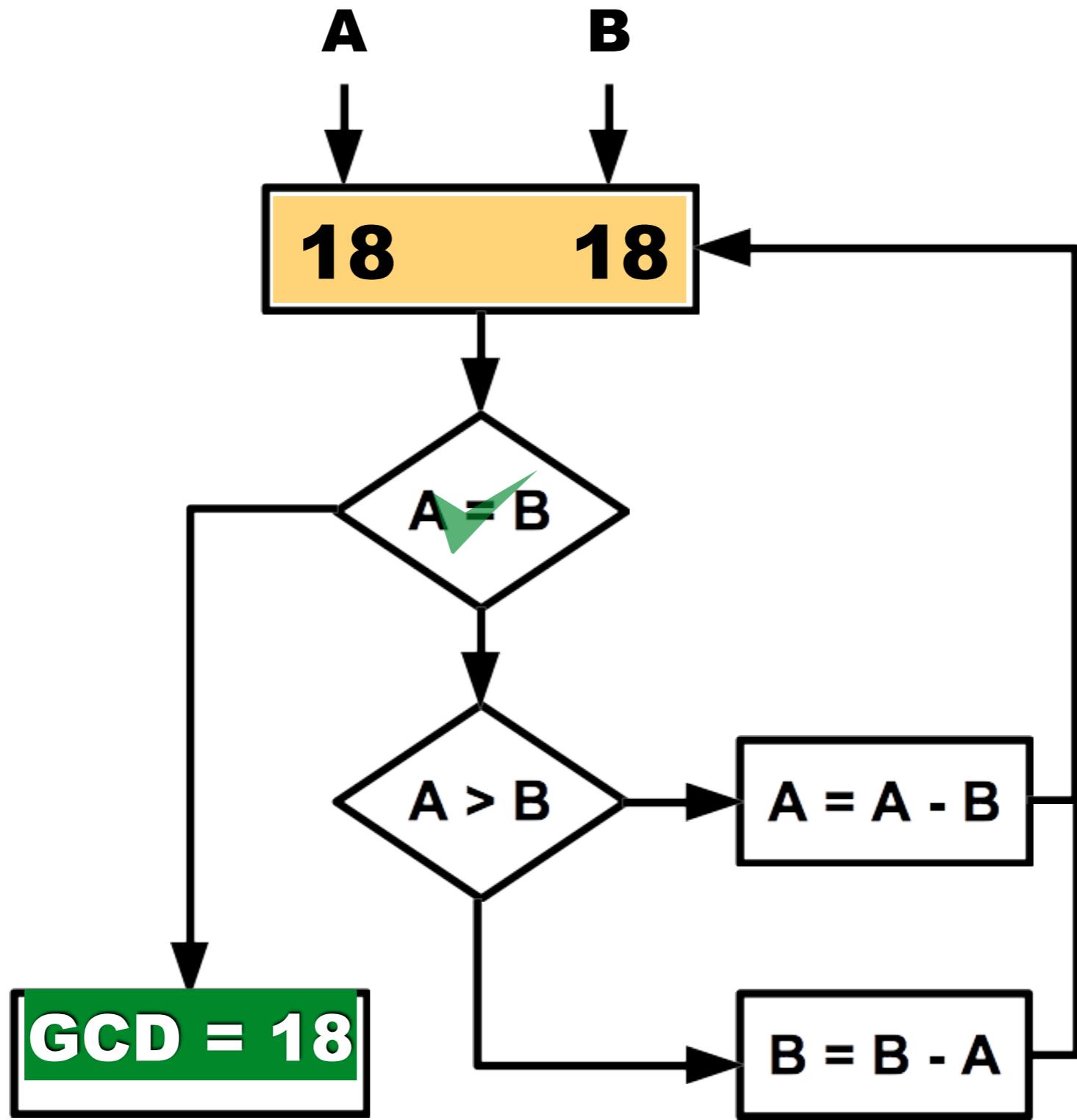
Euclidean Algorithm

36, 54

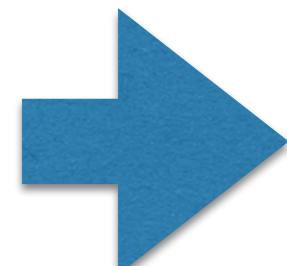
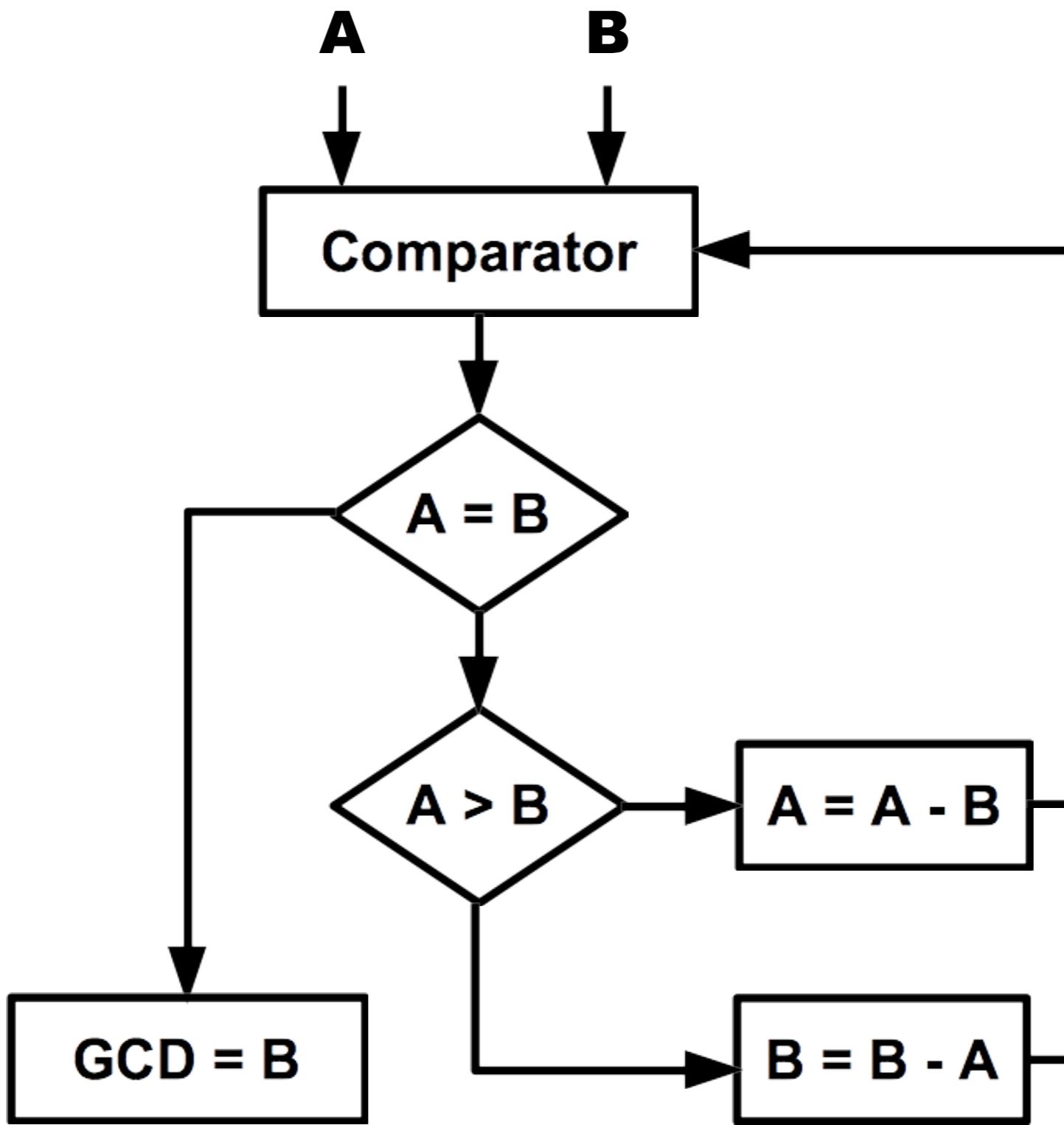


Euclidean Algorithm

36, 54

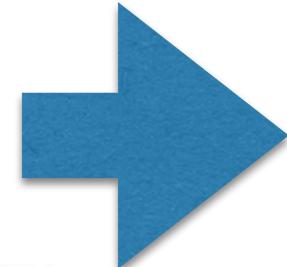
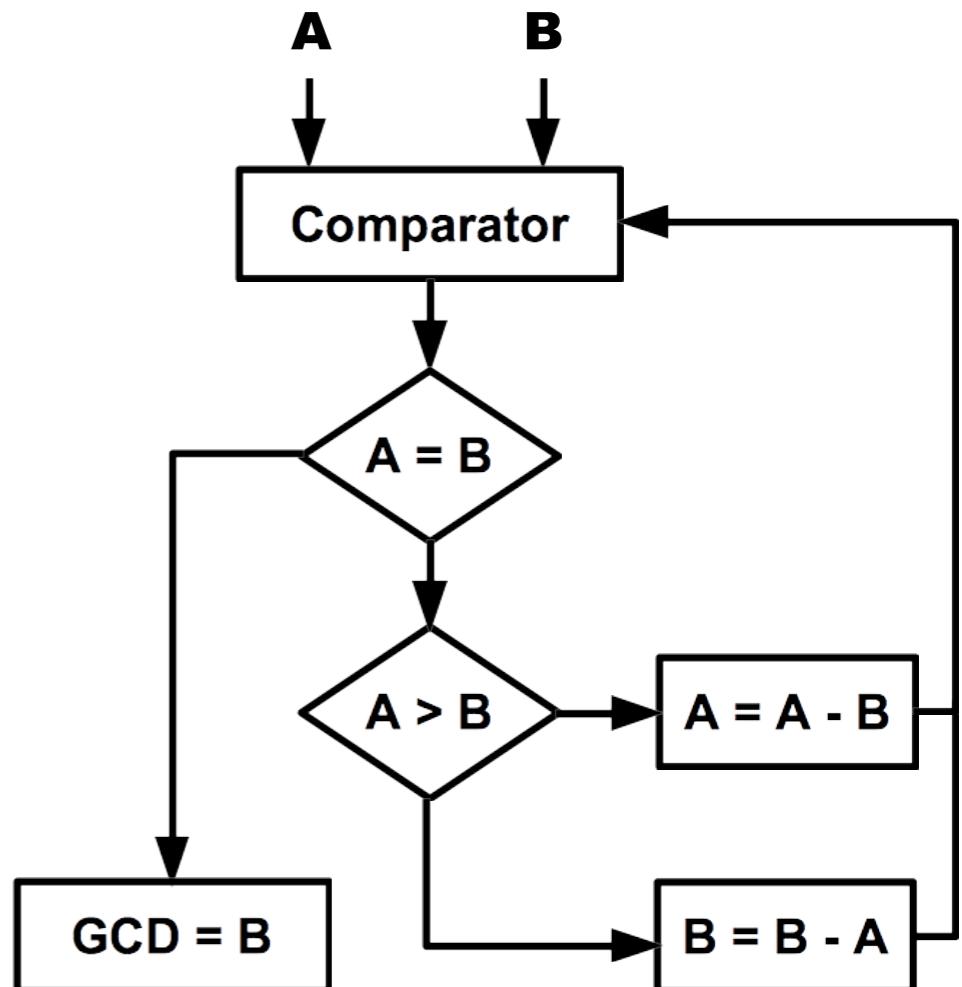


Behavioural Model (Loop)



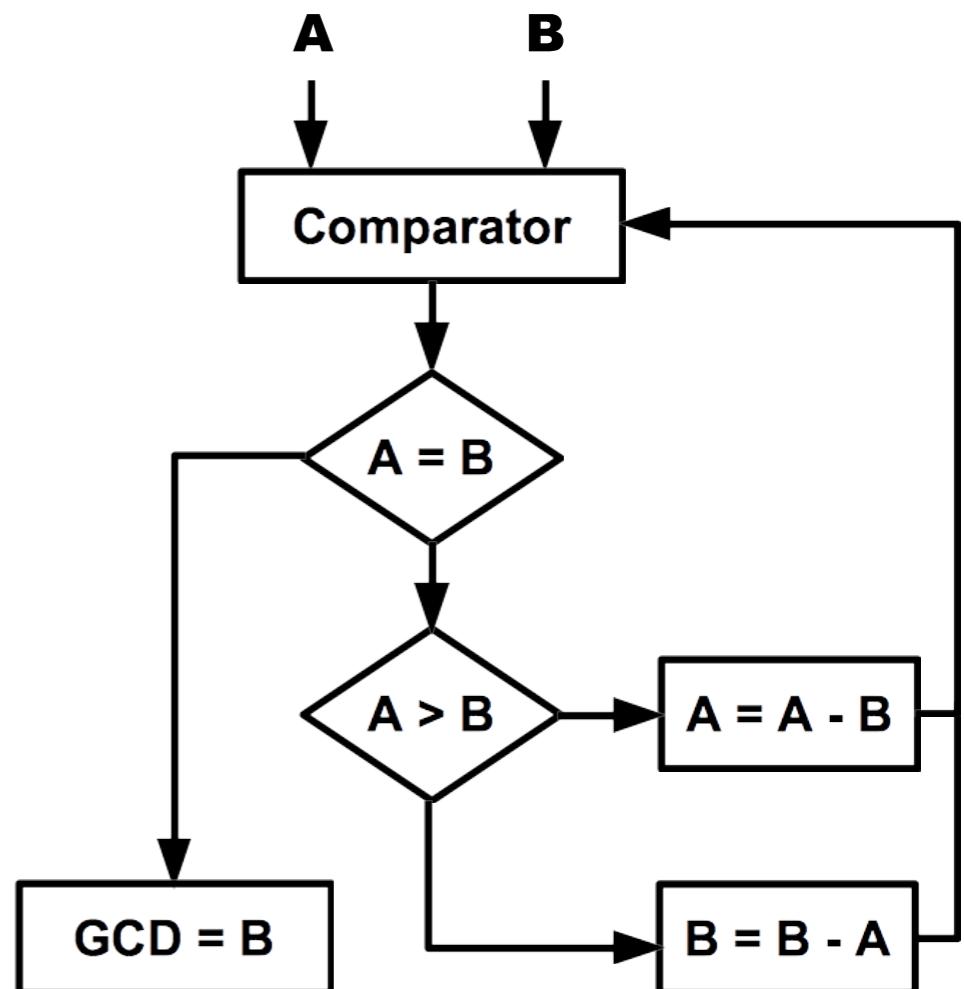
Behavioural Model (Loop)

```
Process (A, B)
Variable AX, BX: Signed (15 downto 0);
Begin
    AX := A;
    BX := B;
    While (AX /= BX) Loop
        If (AX > BX) Then
            AX := AX - BX;
        Else
            BX := BX - AX;
        End If;
    End Loop;
    GCD <= BX;
End Process;
```



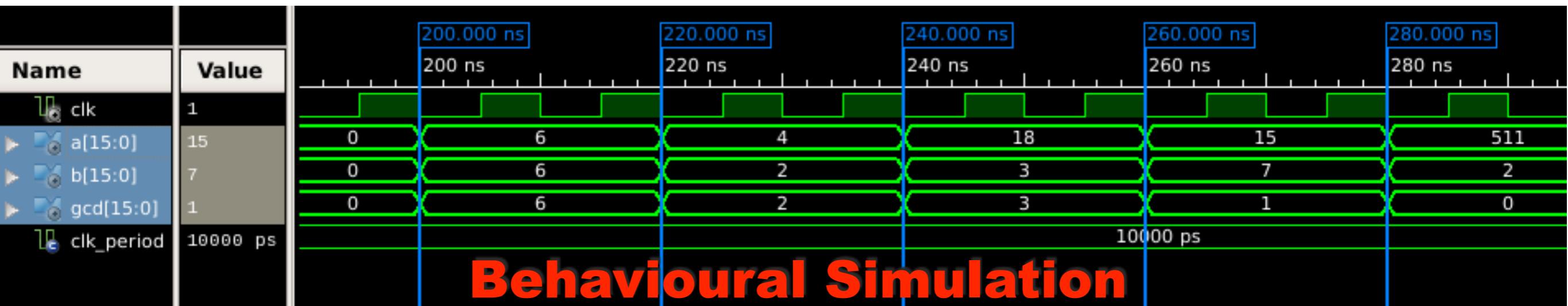
Behavioural Model (Loop)

```
Process (A, B)
Variable AX, BX: Signed (N-1 downto 0);
Begin
    AX := A;
    BX := B;
    for i in 1 to 100 Loop
        If (AX /= BX) Then
            If (AX > BX) Then
                AX := AX - BX;
            Else
                BX := BX - AX;
            End If;
        Else
            GCD <= BX;
        End If;
    End Loop;
End Process;
```

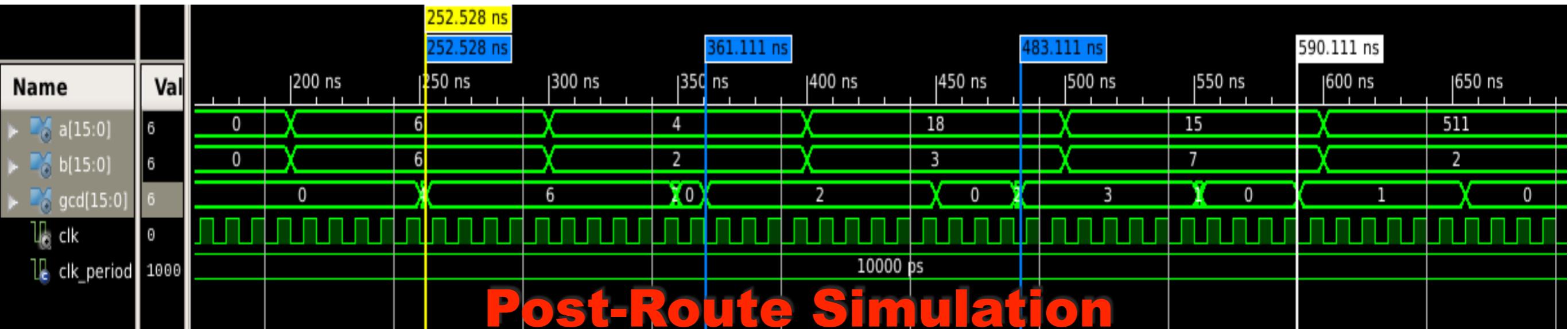


For Loop (Simulation)

Behavioural Simulation

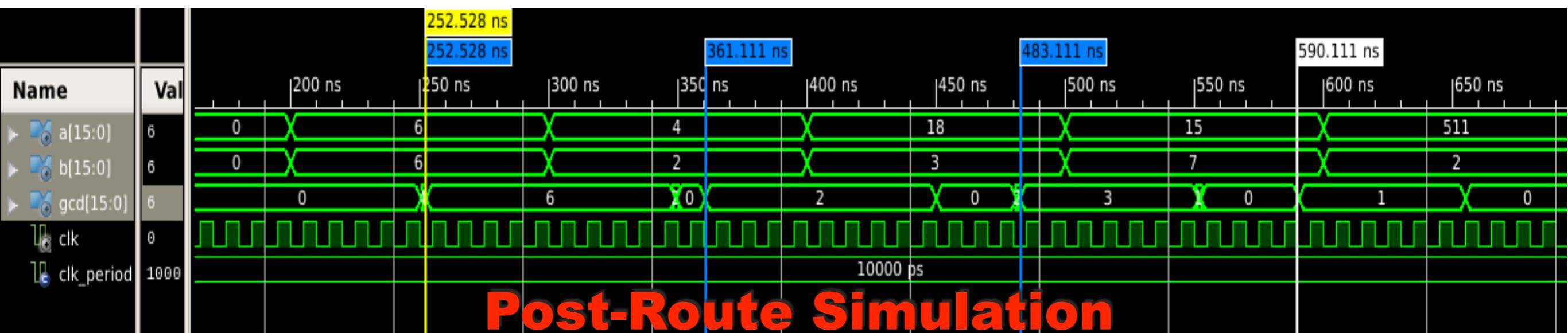
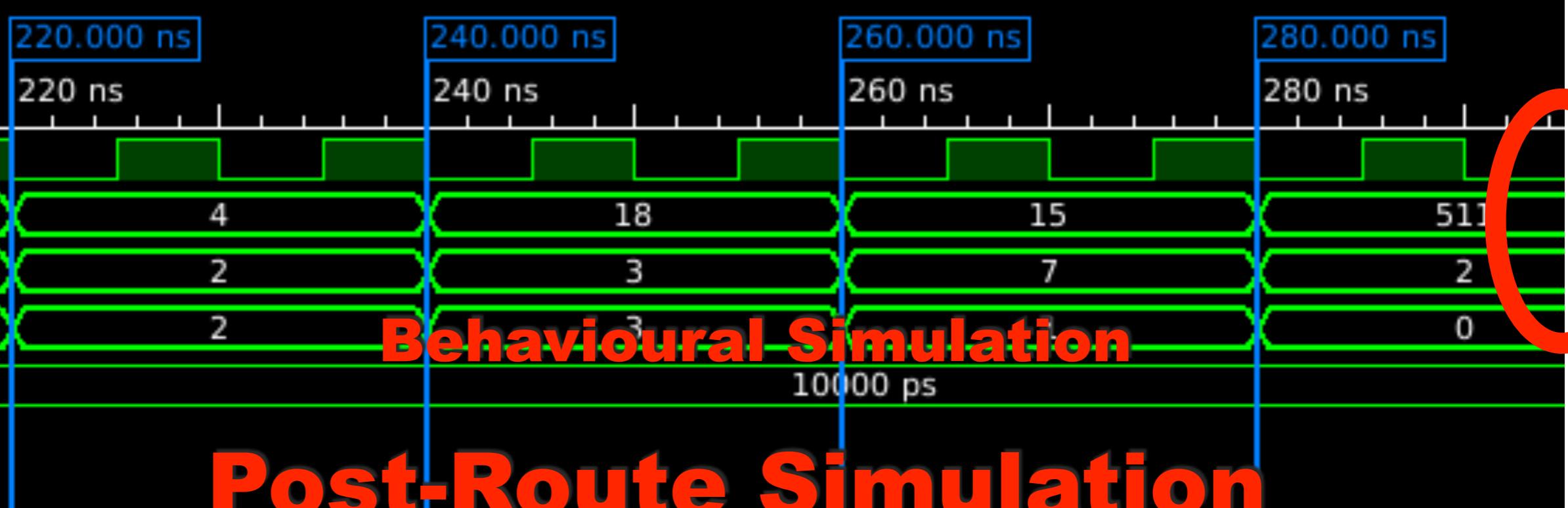


Post-Route Simulation



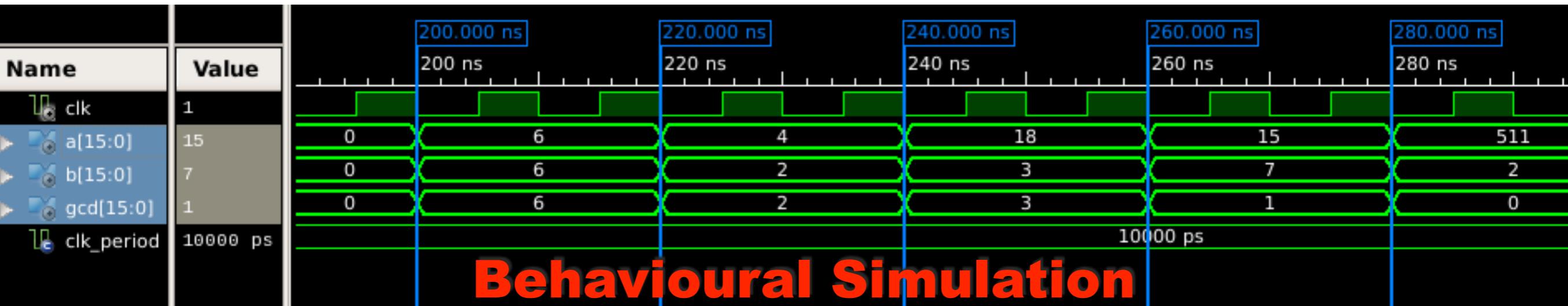
For Loop (Simulation)

Behavioural Simulation

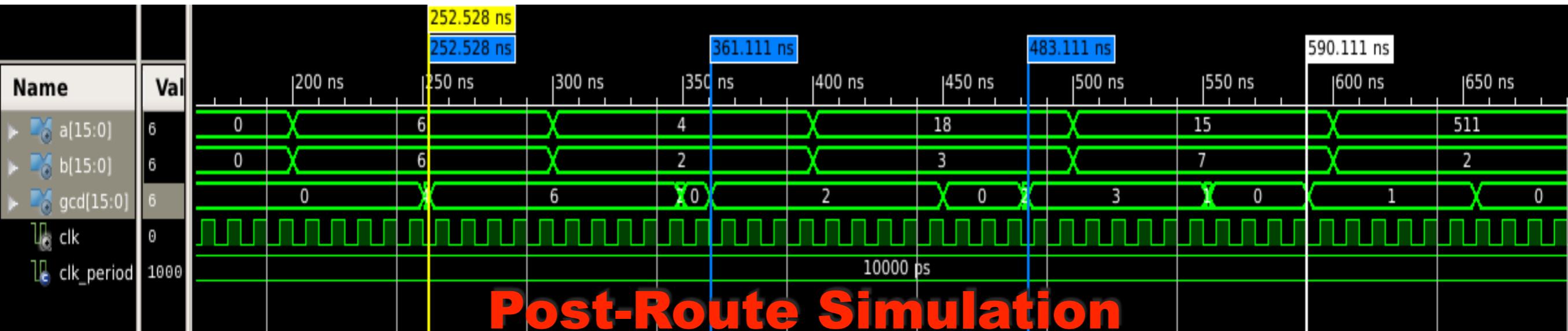


For Loop (Simulation)

Behavioural Simulation

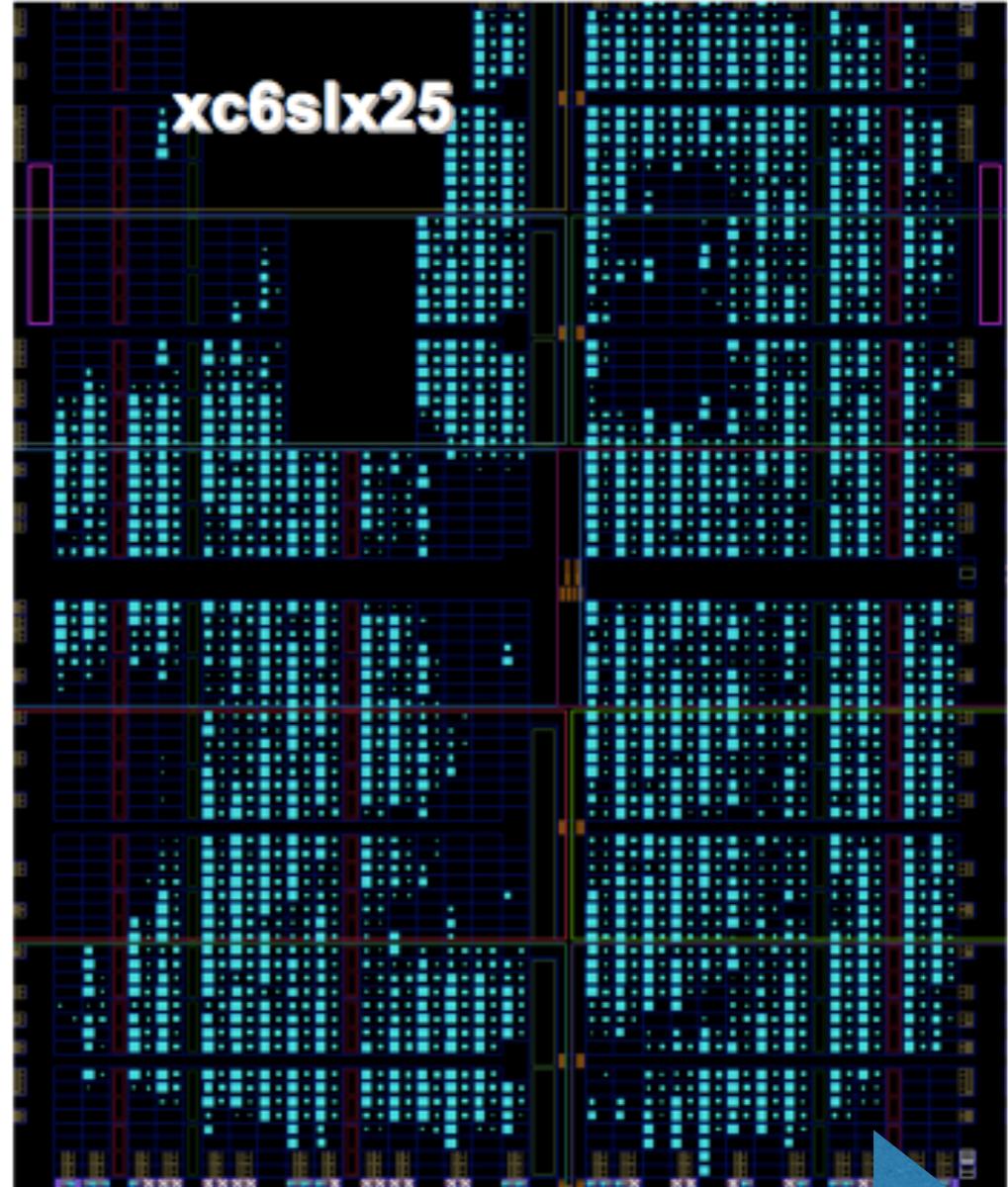
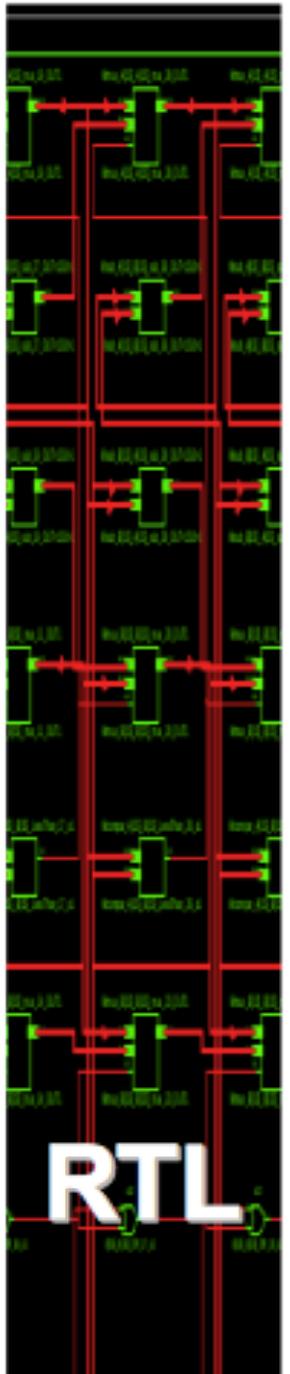


Post-Route Simulation



For Loop (RTL, Implementation)

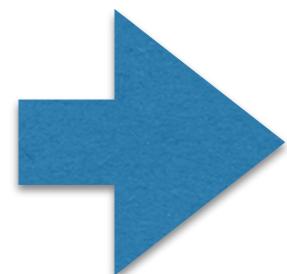
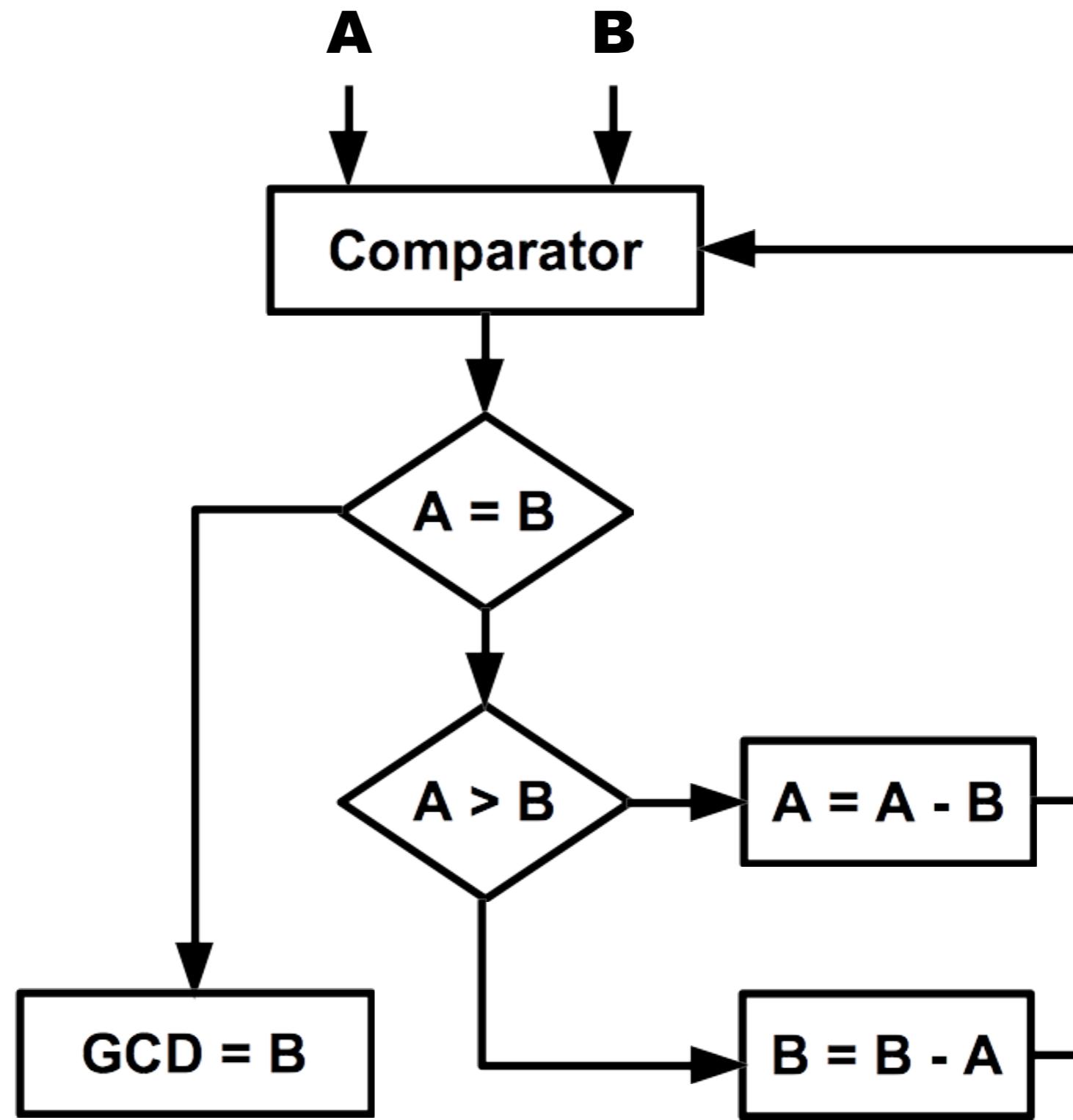
RTL



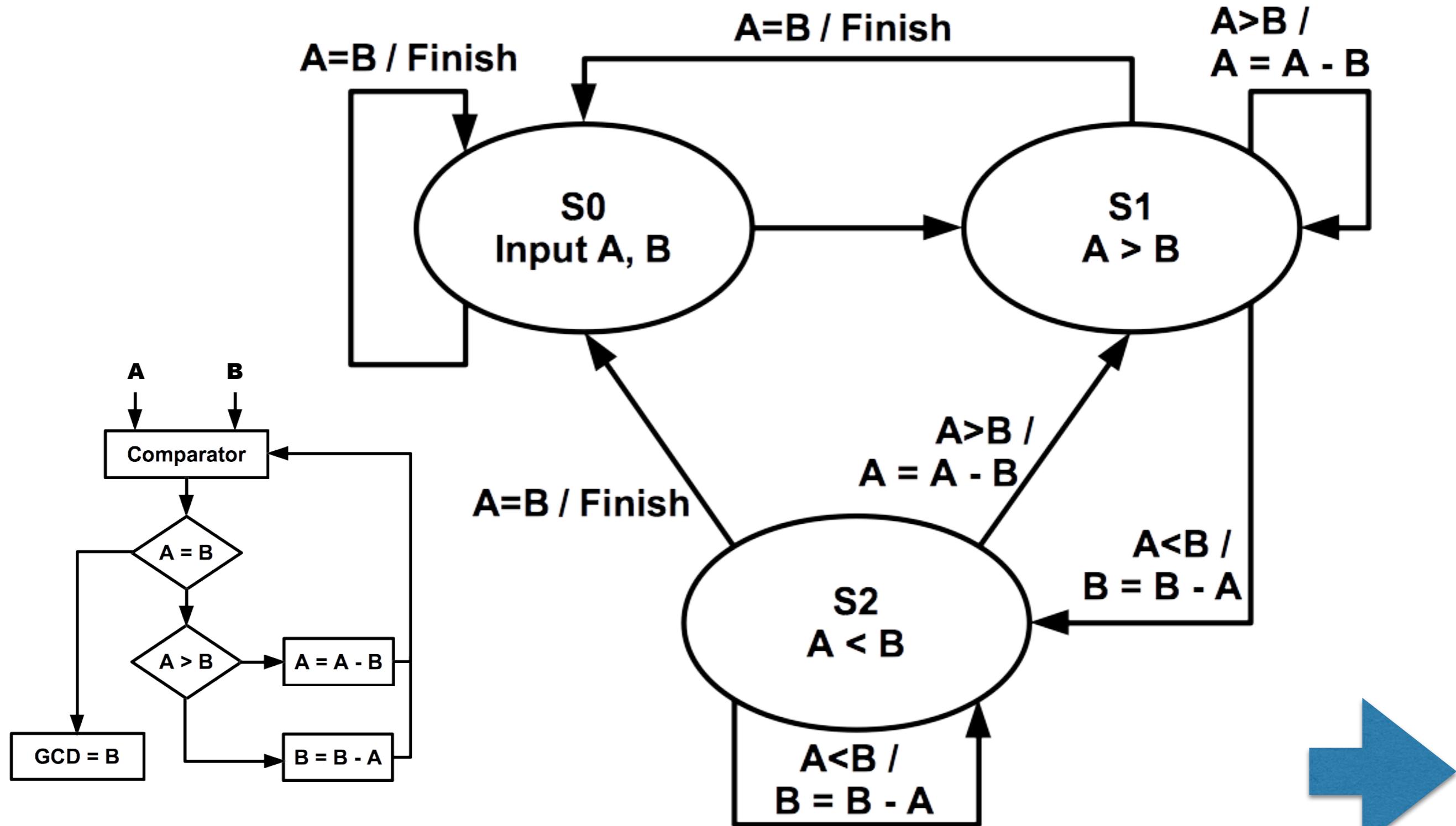
For Loop (RTL, Implementation)

Device	#	# Bits	Note
# Adders/Subtractors	198	16	-
# Registers	0	-	-
# Comparators	200	16	"100= & 99<"
# Multiplexers	298	16	2-1
# FSM	0	-	-
# DSP	0	-	-
# XOR	0	-	-
Time Delay	478		
	#	out of	%
# Slices	2748	3758	73
# LUTs	8776	15032	58
# MUXCYs	4760	7516	63
# Registers	0	30064	0

Register Transfer Model (SM)



Register Transfer Model (SM)

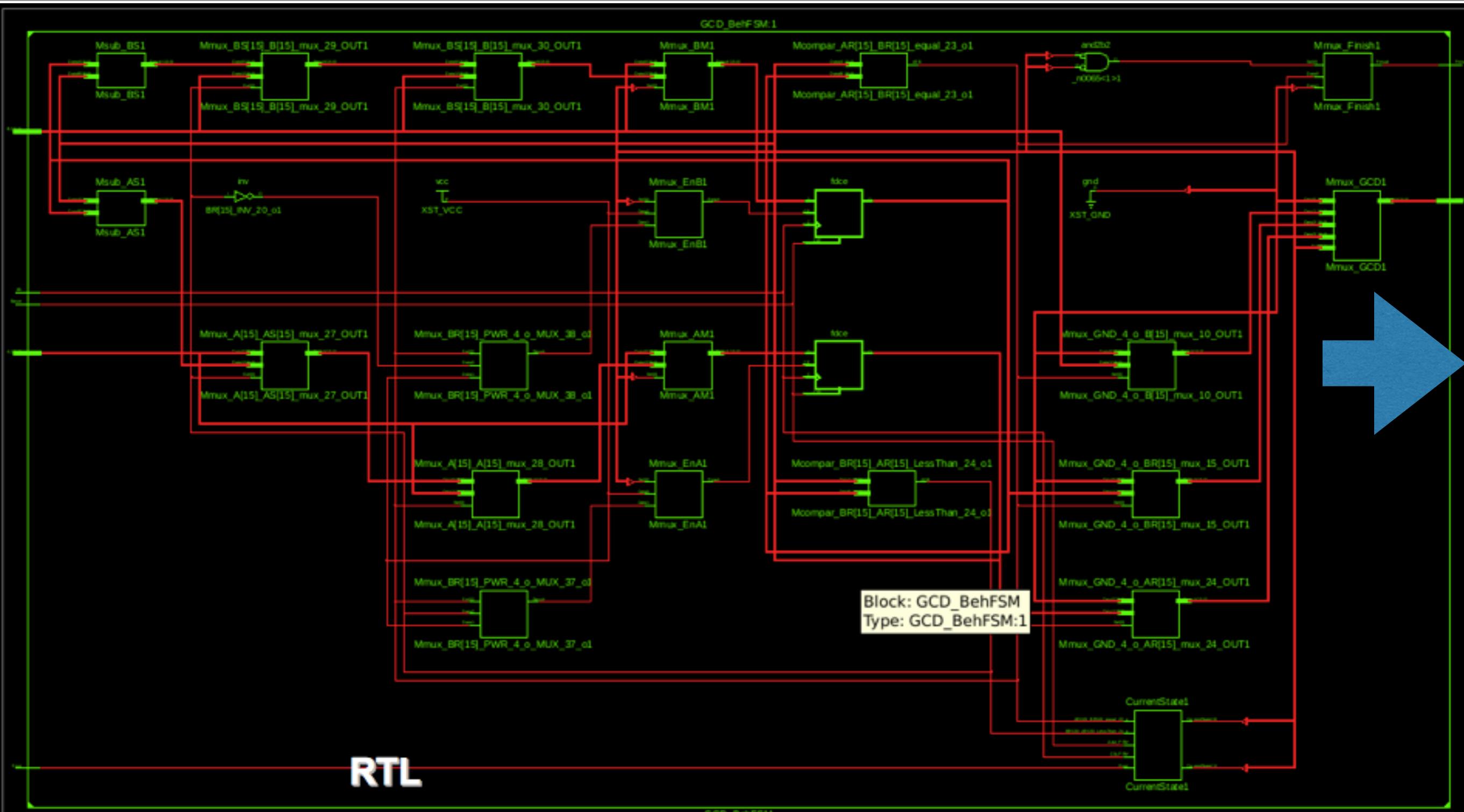


ASM (Code Sample)

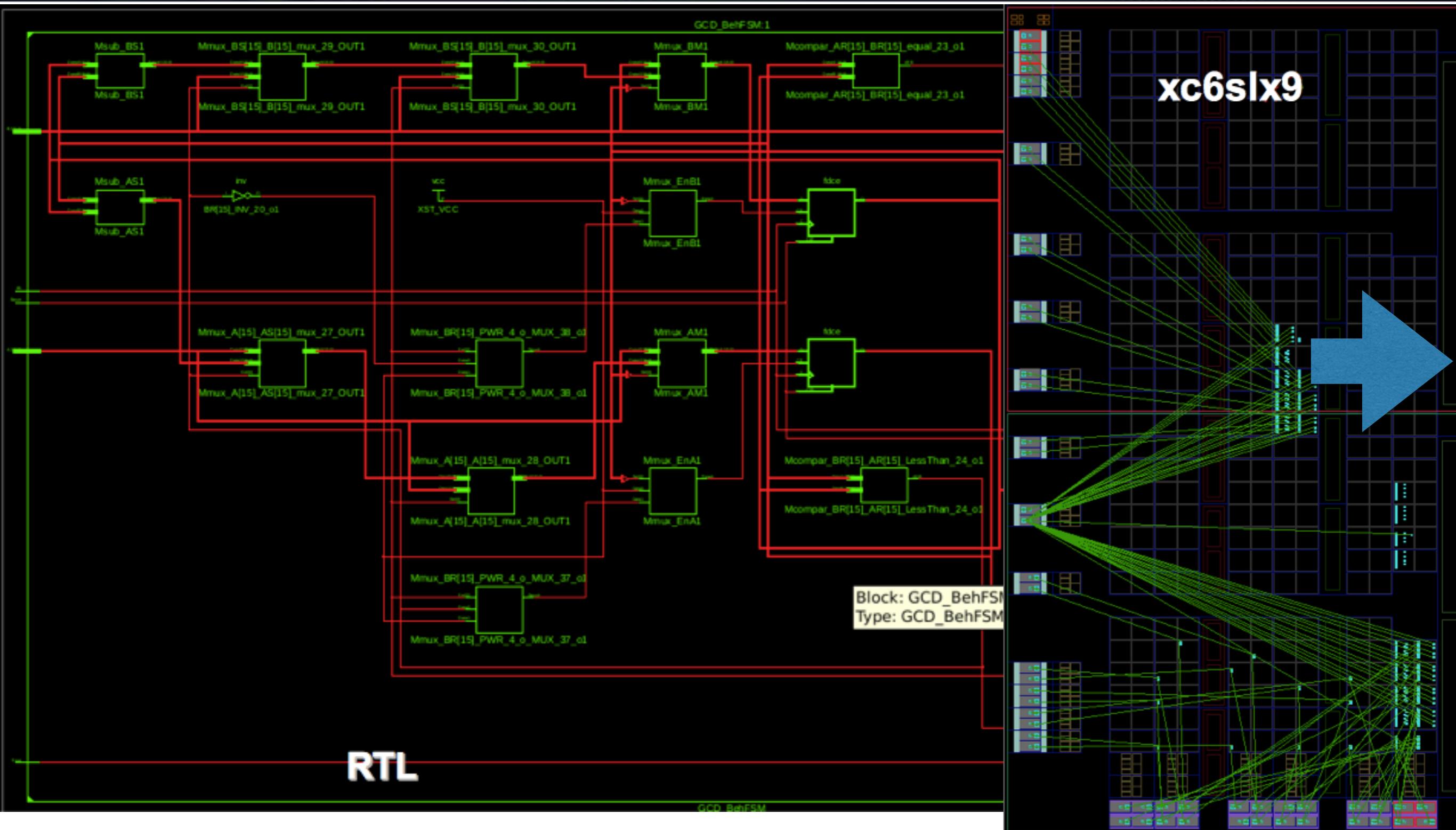
```
AS <= AR - BR;  
BS <= BR - AR;  
  
WHEN S1 =>  
  IF (AR = BR) THEN  
    Tmp_GCD <= B;  
    Finish <= '1';  
    NextState <= S0;  
  ELSIF (AR > BR) THEN  
    AM <= AS;  
    EnA <= '1';  
    EnB <= '0';  
    NextState <= S2;  
  ELSE  
    NextState <= S2;  
END IF;
```

```
WHEN S2 =>  
  IF (AR = BR) THEN  
    Tmp_GCD <= BR;  
    Finish <= '1';  
    NextState <= S0;  
  ELSIF (AR > BR) THEN  
    AM <= AS;  
    EnA <= '1';  
    EnB <= '0';  
    NextState <= S2;  
  ELSE  
    BM <= BS;  
    EnA <= '0';  
    EnB <= '1';  
    NextState <= S2;  
  END IF;
```

ASM (Implementation)



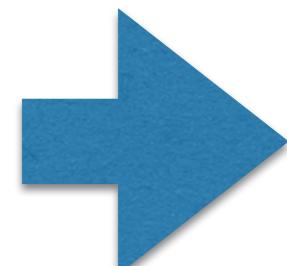
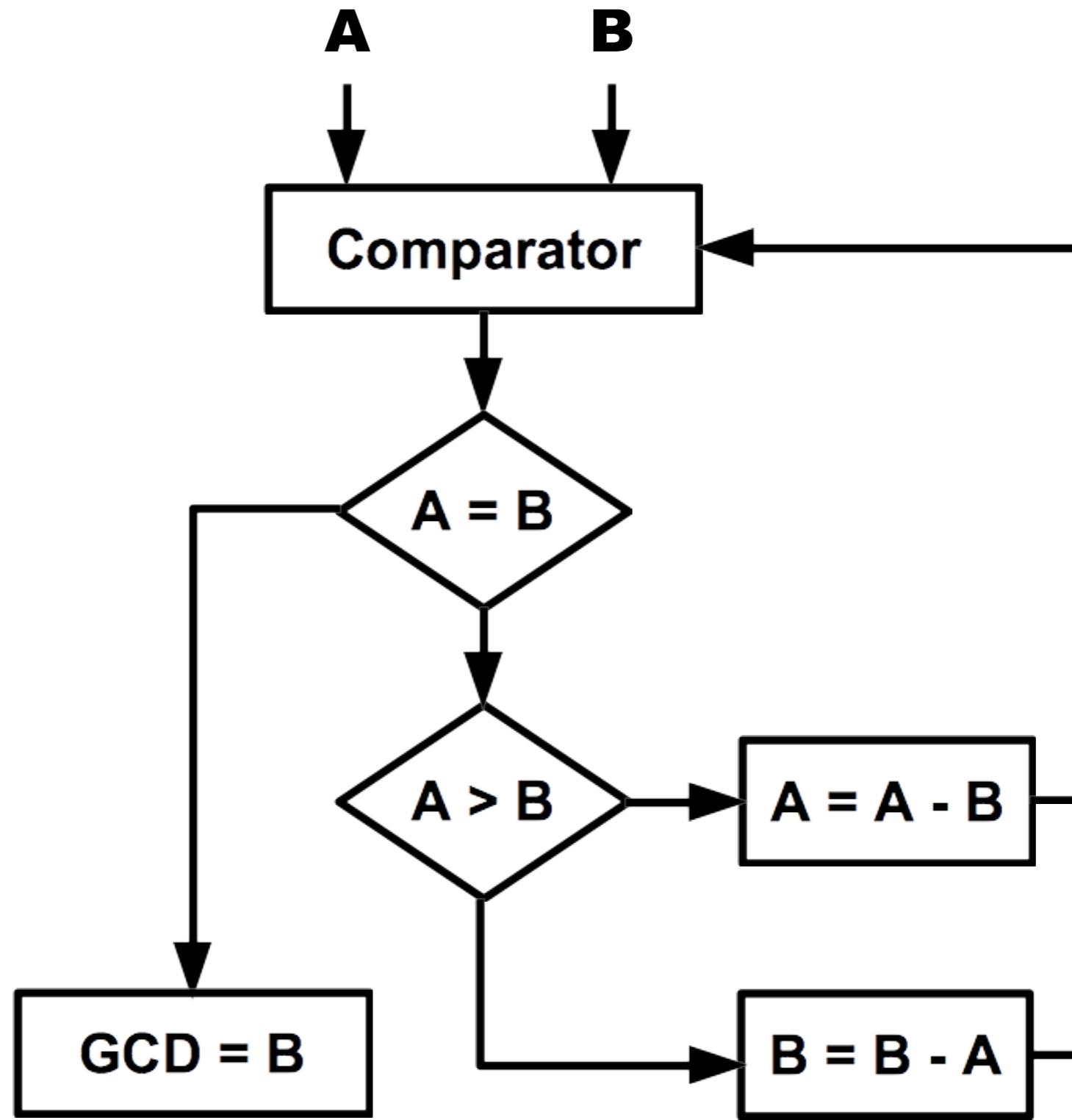
ASM (Implementation)



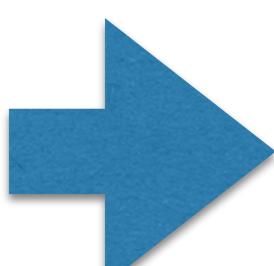
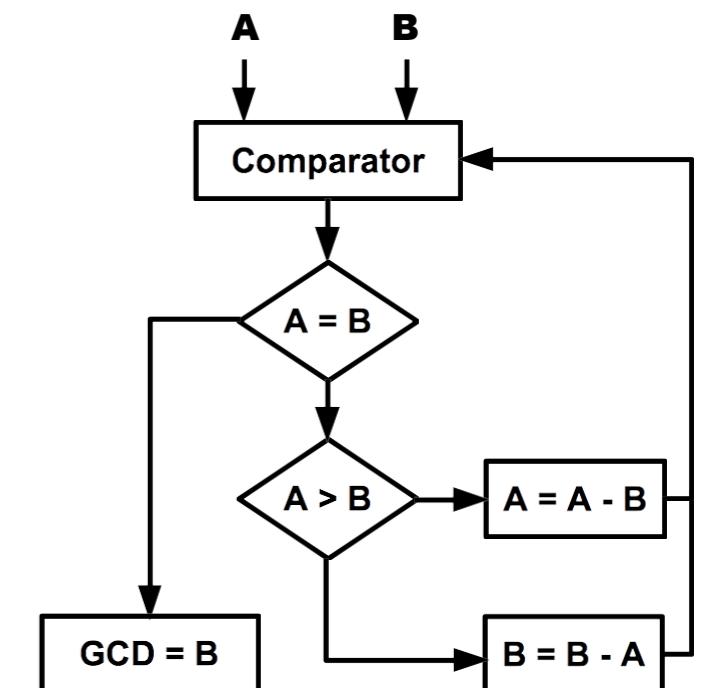
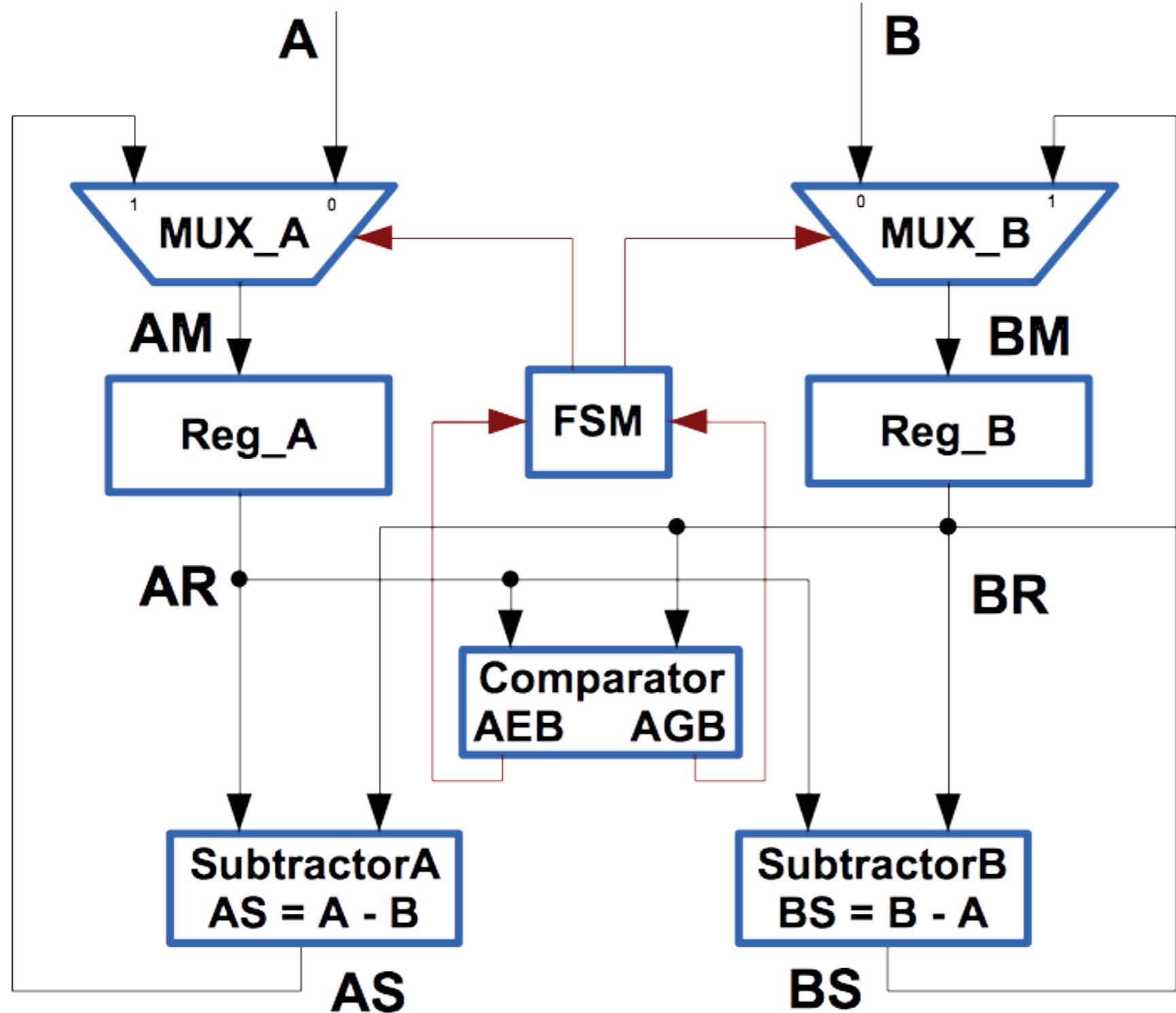
ASM (Implementation)

Device	#	# Bits	Note
# Adders/Subtractors	2	16	
# Registers	32	1	FF
# Comparators	2	16	"= & <"
# Multiplexers	14	16 (8,1), 1(5)	2-1(13), 3-1(1)
# FSM	1	-	-
# DSP	0	-	-
# XOR	0	-	-
Time Delay		20.7	
			
	#	out of	%
# Slices	36	3758	1
# LUTs	98	15032	1
# MUXCYs	48	7516	1
# Registers	34	30064	1

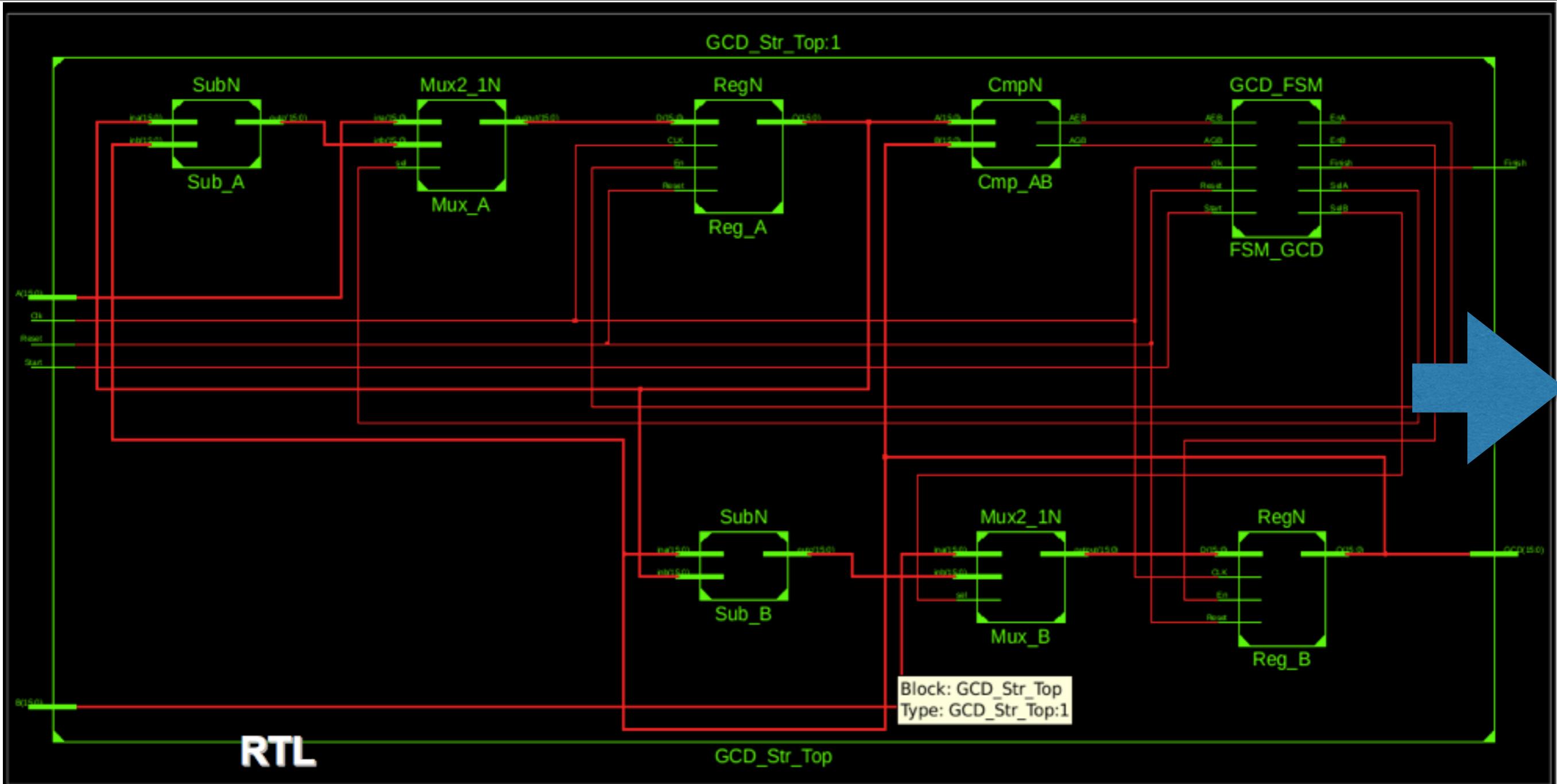
Direct Structural Level



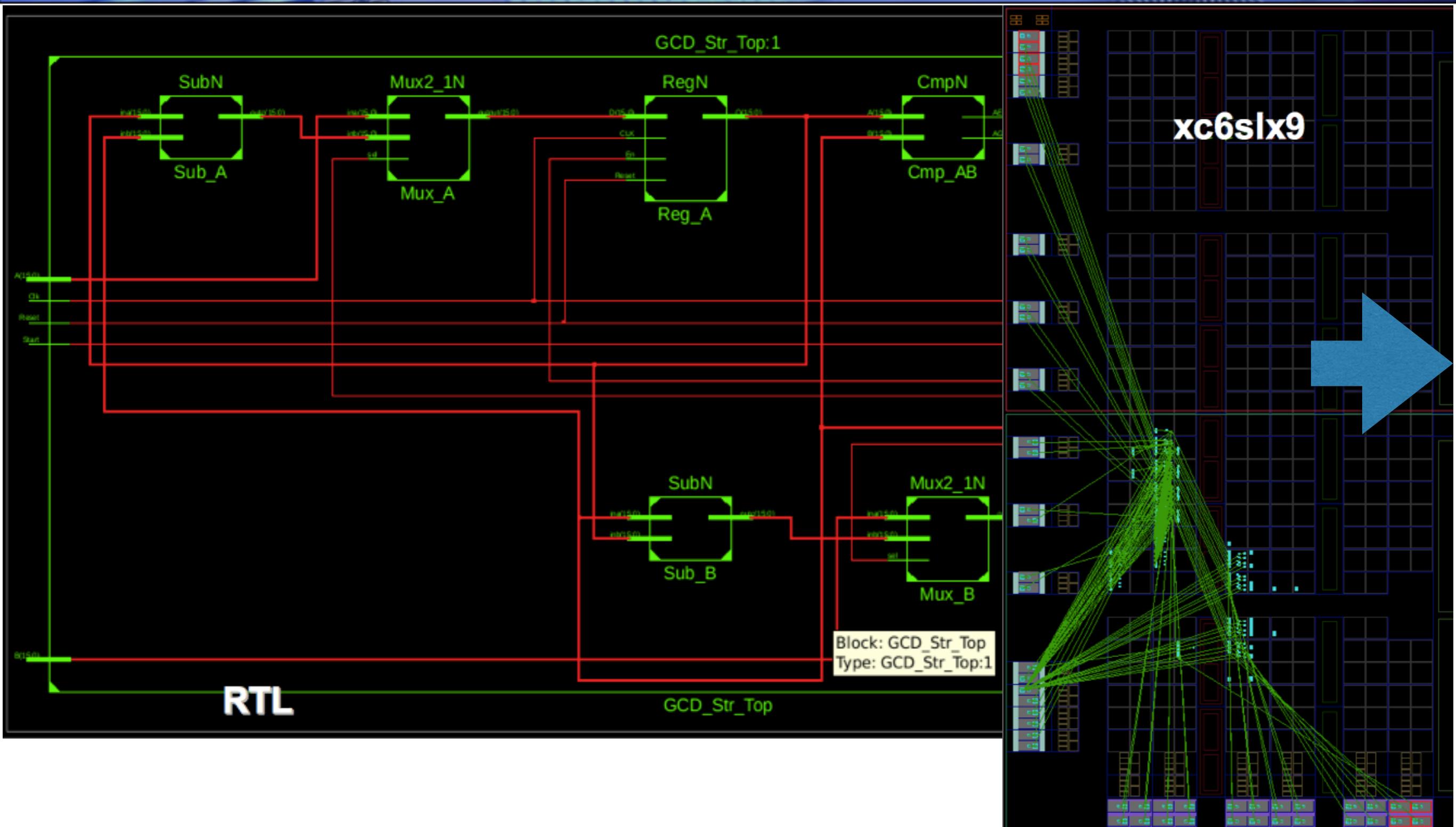
Direct Structural Level



Dir. Structural (Implementation)



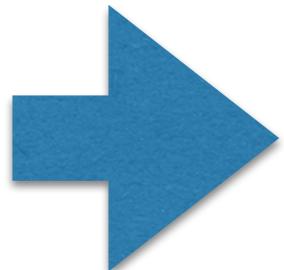
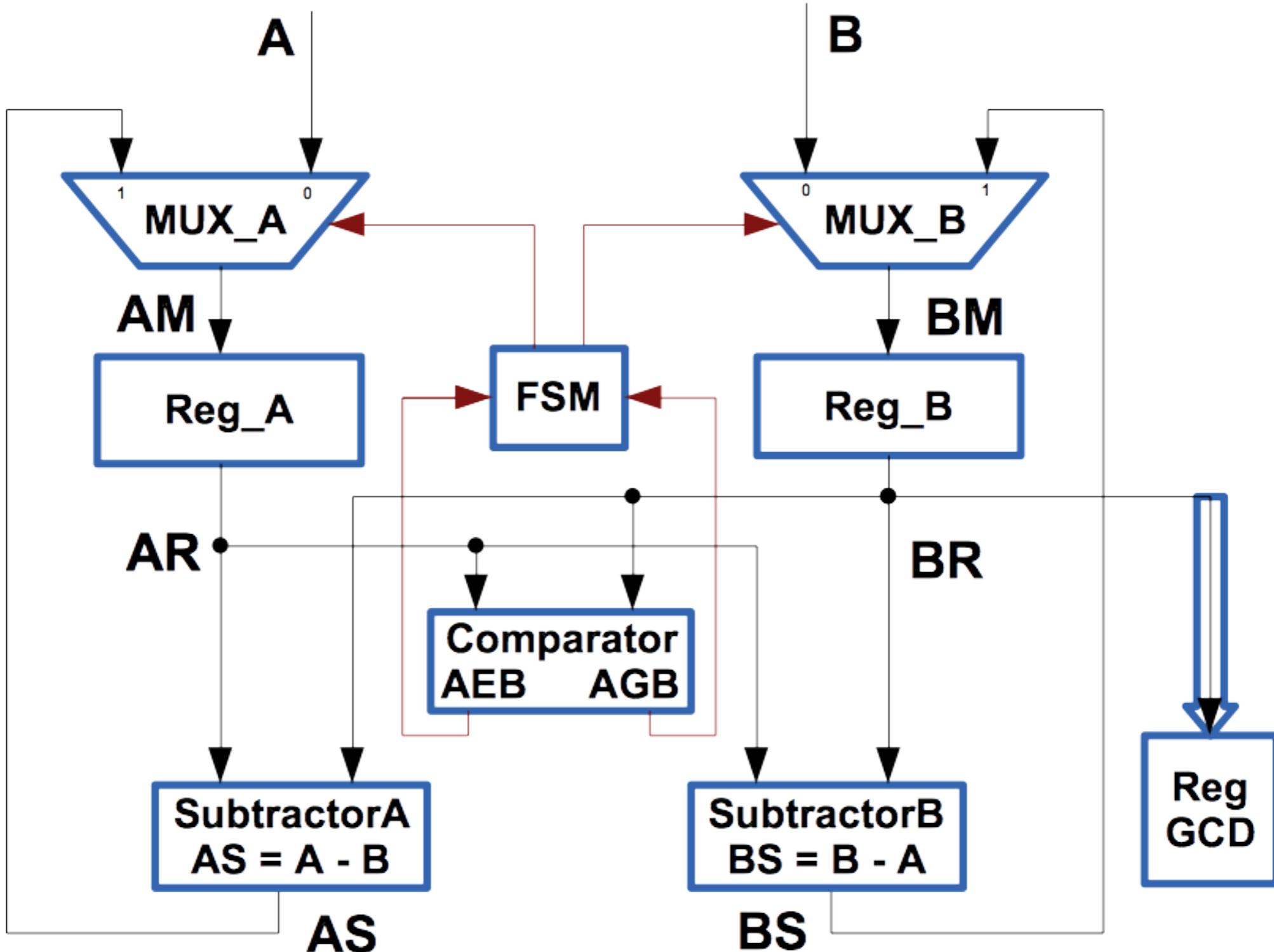
Dir. Structural (Implementation)



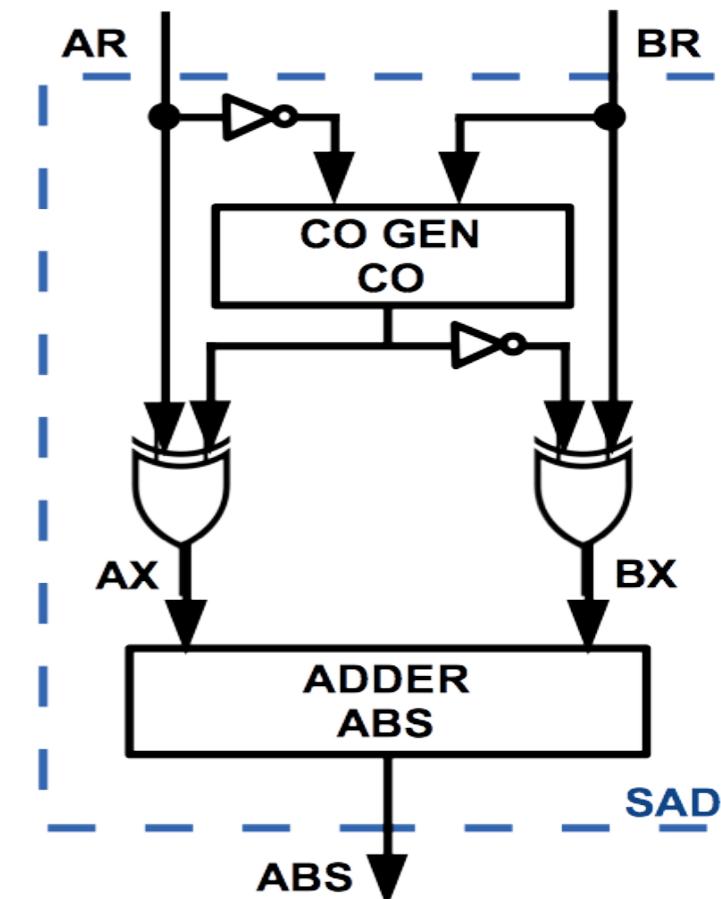
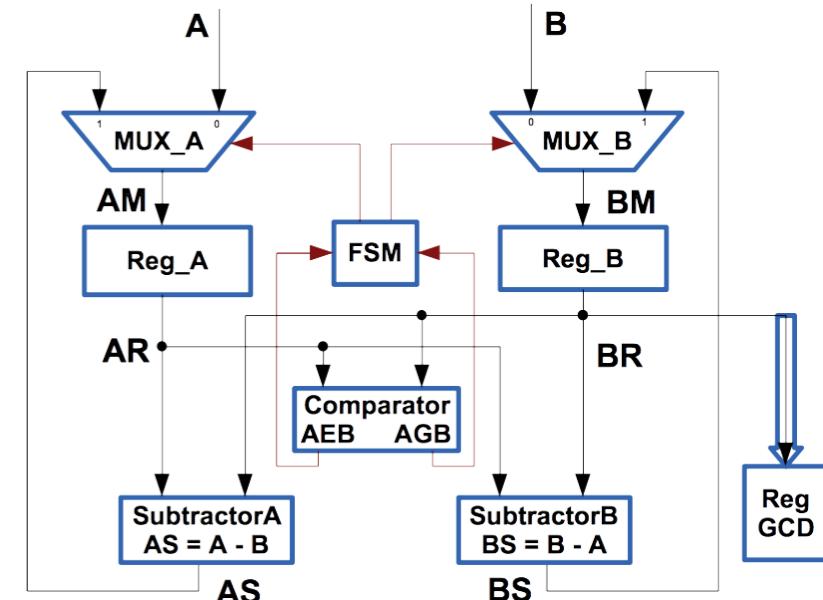
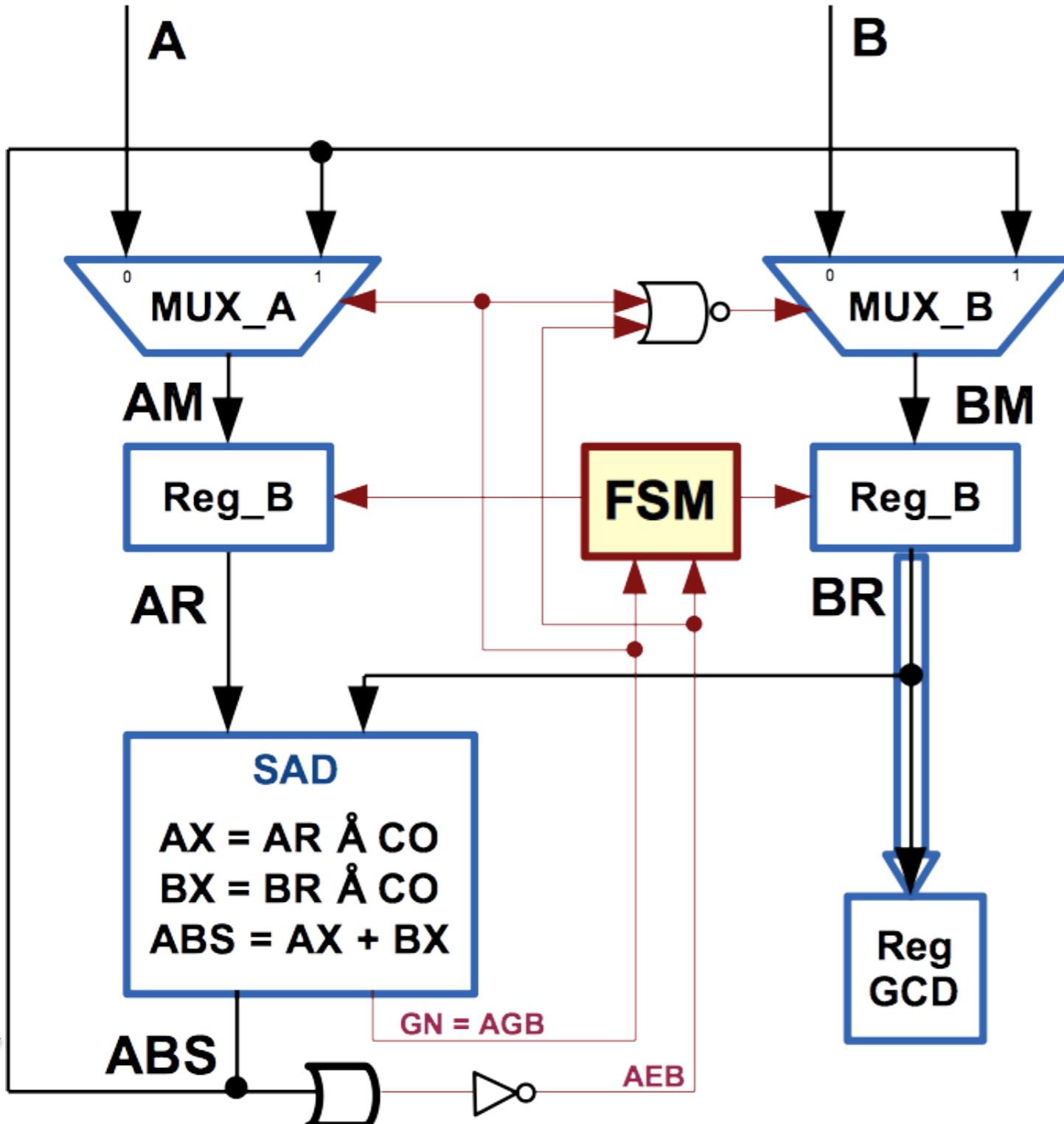
Dir. Structural (Implementation)

Device	#	# Bits	Note
# Adders/Subtractors	2	16	Sub/Accum.
# Registers	2	16	32 FF
# Comparators	2	16	"= & <"
# Multiplexers	11	2 (16), 9 (1)	2-1
# FSM	1	-	-
# DSP	0	-	-
# XOR	0	-	-
Time Delay		15.5	
	#	out of	%
# Slices	30	3758	1
# LUTs	85	15032	1
# MUXCYs	52	7516	1
# Registers	34	30064	1

Optimized Structural Level



Optimized Structural Level



Opt. Str. (Code Sample)

```
BN <= NOT B;
P <= A xor BN;
G <= A AND BN;

GP4: FOR i IN 1 TO (N/4) Generate -- 4-BIT BLOCK
P4(i) <= P(4*i-1) AND (P(4*i-2) AND P(4*i-3) AND P(4*i-4));
G4(i) <= G(4*i-1) OR (G(4*i-2) AND P(4*i-1)) OR
(G(4*i-3) AND P(4*i-1) AND P(4*i-2)) OR
(G(4*i-4) AND P(4*i-1) AND P(4*i-2) AND P(4*i-3));
END Generate GP4;

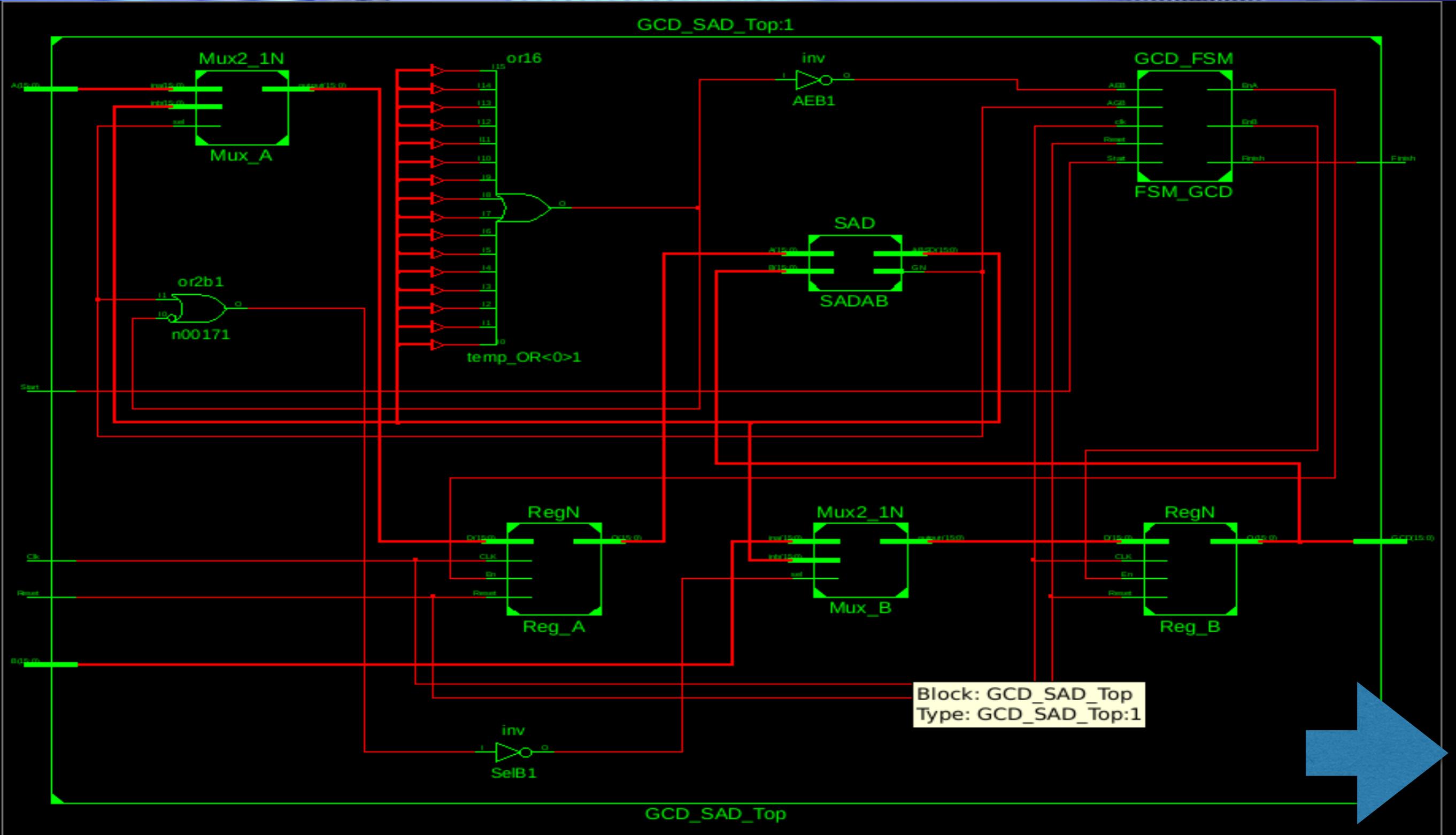
-- 16-BIT BLOCK
GN <= G4(4) OR (G4(3) AND P4(4)) OR
(G4(2) AND P4(4) AND P4(3)) OR
(G4(1) AND P4(4) AND P4(3) AND P4(2));
-- GN = '1', WHEN A>B!

ABXOR: FOR i IN 0 TO N-1 Generate
AX(i) <= A(i) XNOR GN;
BX(i) <= B(i) XOR GN;
END Generate ABXOR;
```

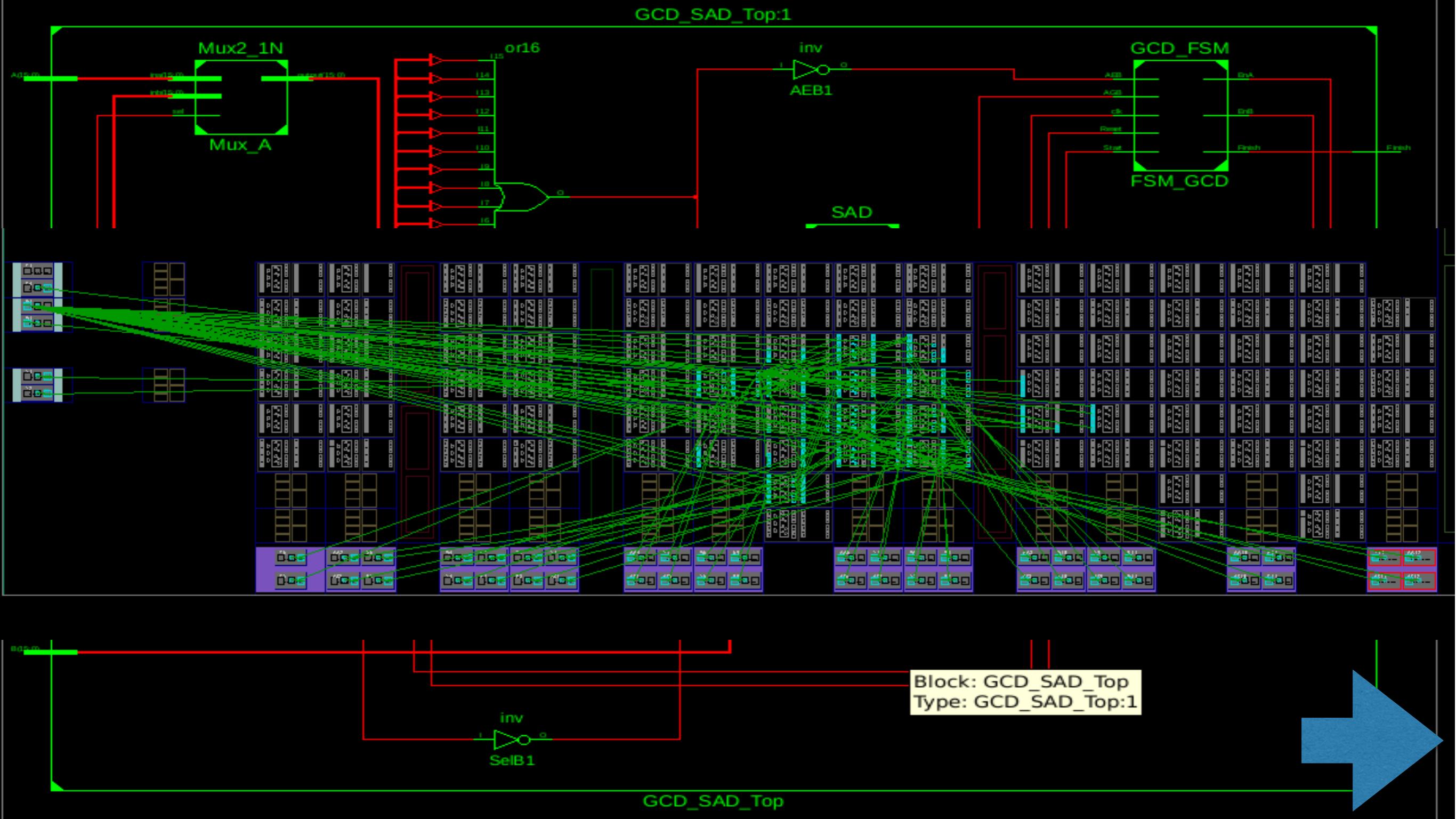
Opt. Str. (Code Sample)

```
Process (ABSD)
    variable temp_OR : STD_LOGIC;
begin
    temp_OR := ABSD(0);
    for i in 1 to 15 loop
        temp_OR := temp_OR OR ABSD(i);
    end loop;
    AEB <= NOT temp_OR;
end process;
AGB <= GN;
-- SelA <= AGB;
SelB <= AGB NOR AEB;
```

Opt. Structural (Implementation)



Opt. Structural (Implementation)



Opt. Structural (Implementation)

Device	#	# Bits	Note
# Adders/Subtractors	1	16	Carry in
# Registers	2	16	32 FF
# Comparators	0		
# Multiplexers	7	2 (16), 5 (1)	2-1
# FSM	1		
# DSP	0		
# XOR	33	1 (16), 32 (1)	
Time Delay		23.6	
	#	out of	%
# Slices	33	3758	1
# LUTs	110	15032	1
# MUXCYs	16	7516	1
# Registers	34	30064	1

Summary

Device	4Loop	ASM	2Sub	SADGCD
# Adders/Subtractors	198	2	2	1
# Registers	0	32	2	2
# Comparators	200	2	2	0
# Multiplexers	298	14	11	7
# FSM	0	1	1	1
# DSP	0	0	0	0
# XOR	0	0	0	33
Time Delay	478	20.7	15.5	23.6

	out of	4Loop	%	ASM	2Sub	SADGCD
# Slices	3758	2748	73	36	30	33
# LUTs	15032	8776	58	98	85	110
# MUXCYs	7516	4760	63	48	52	16
# Registers	30064	0	0	34	34	34

Conclusion & Future Work

- The Design of 16-bit GCD and its Implementation it on Xilinx Spartan6 FPGA was demonstrated.
- The area-speed tradeoff has been shown in different Levels:
 - High Behavioural Level
 - ASM Behavioural Level
 - Direct Structural Level
 - Optimized Structural Level
- Future Work: Primitives + Check new FPGAs where GCD might be implemented inside a DSP

References

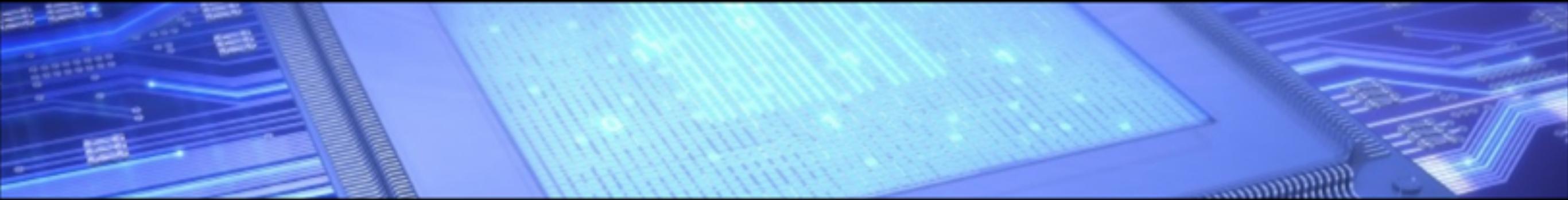
<https://www.youtube.com/watch?v=DMSaYhD1GkM>

<http://esd.cs.ucr.edu/labs/tutorial/>

http://en.wikipedia.org/wiki/Greatest_common_divisor

[And many others](#)

[Spartan-6 FPGA Configurable Logic Block User Guide \(UG384 \(v1.1\) February 23, 2010\)](#)



Thank You