




A	B	\bar{A}	$A+B$	$A \cdot B$	$\overline{A+B}$	$\overline{A \cdot B}$	$A \oplus B$
0	0	1	0	0	1	1	0
0	1	1	1	0	0	1	1
1	0	0	1	0	0	1	1
1	1	0	1	1	0	0	0

NOT OR AND NOR NAND XOR

		NOT		$A \rightarrow L$	<table border="1"><tr><th>A</th><th>L</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	L	0	1	1	0																									
A	L																																			
0	1																																			
1	0																																			
<table border="1"><tr><th>A</th><th>B</th><th>L</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	L	0	0	0	0	1	0	1	0	0	1	1	1			NAND		$A, B \rightarrow L$	<table border="1"><tr><th>A</th><th>B</th><th>L</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	L	0	0	1	0	1	1	1	0	1	1	1	0
A	B	L																																		
0	0	0																																		
0	1	0																																		
1	0	0																																		
1	1	1																																		
A	B	L																																		
0	0	1																																		
0	1	1																																		
1	0	1																																		
1	1	0																																		
<table border="1"><tr><th>A</th><th>B</th><th>L</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	L	0	0	0	0	1	1	1	0	1	1	1	1			NOR		$A, B \rightarrow L$	<table border="1"><tr><th>A</th><th>B</th><th>L</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	L	0	0	1	0	1	0	1	0	0	1	1	0
A	B	L																																		
0	0	0																																		
0	1	1																																		
1	0	1																																		
1	1	1																																		
A	B	L																																		
0	0	1																																		
0	1	0																																		
1	0	0																																		
1	1	0																																		

BB=B

ASSEMBLY CODE:

MOV target,source

BRNE k waar k de relatieve jump is en BRNE regel meetelt.

SUB → BRNE alleen als resultaat

Relationship	Dual form	Name
$AB = BA$	$A + B = B + A$	commutative (1)
$1A = A$	$0 + A = A$	identity (2)
$A\bar{A} = 0$	$A + \bar{A} = 1$	complement (3)
$A(B + C) = AB + AC$	$A + BC = (A + B)(A + C)$	distributive (4)
$\overline{AB} = \bar{A} + \bar{B}$	$\overline{A + B} = \bar{A} \cdot \bar{B}$	DeMorgan (5)

SUB niet 0 is

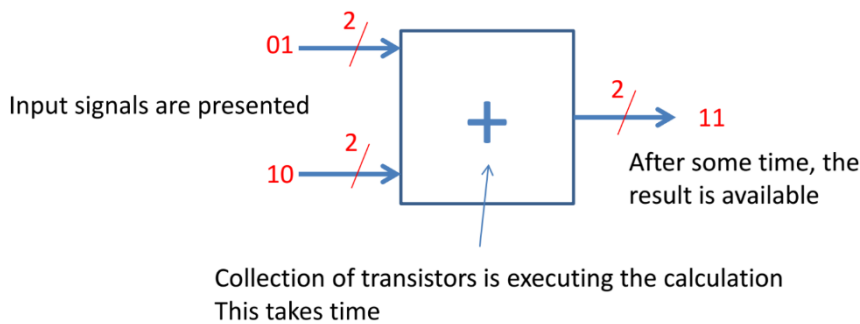
Negative

The three possibilities from the previous slide are:

- A. Use first bit for the minus sign
- B. 1's complement: invert all bits when negative
- C. 2's complement: continue counting after +7 with -8.

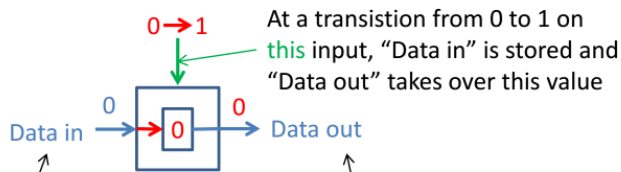
decimal	A	B	C
-8			1000
-7	1111	1000	1001
-6	1110	1001	1010
-5	1101	1010	1011
-4	1100	1011	1100
-3	1011	1100	1101
-2	1010	1101	1110
-1	1001	1110	1111
0	{ 1000	{ 1111	{ 0000
	{ 0000	{ 0000	{ 0000
+1	0001	0001	0001
+2	0010	0010	0010
+3	0011	0011	0011
+4	0100	0100	0100
+5	0101	0101	0101
+6	0110	0110	0110
+7	0111	0111	0111

Combinatorial circuits



In programs operations are coded by 'Mnemonics'. For example '+' = ADD, '*' = MUL.

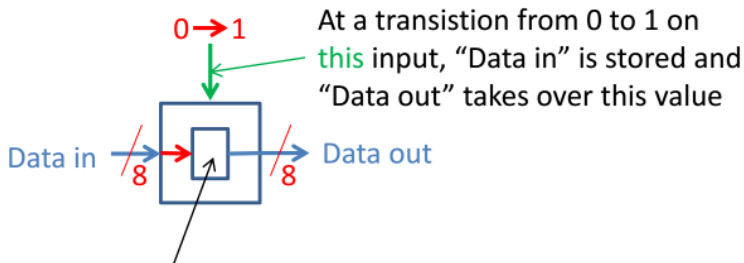
Register



Reasoning backwards: "Data in" comes from
 - Another memory element
 - "outside" the processor (I/O)

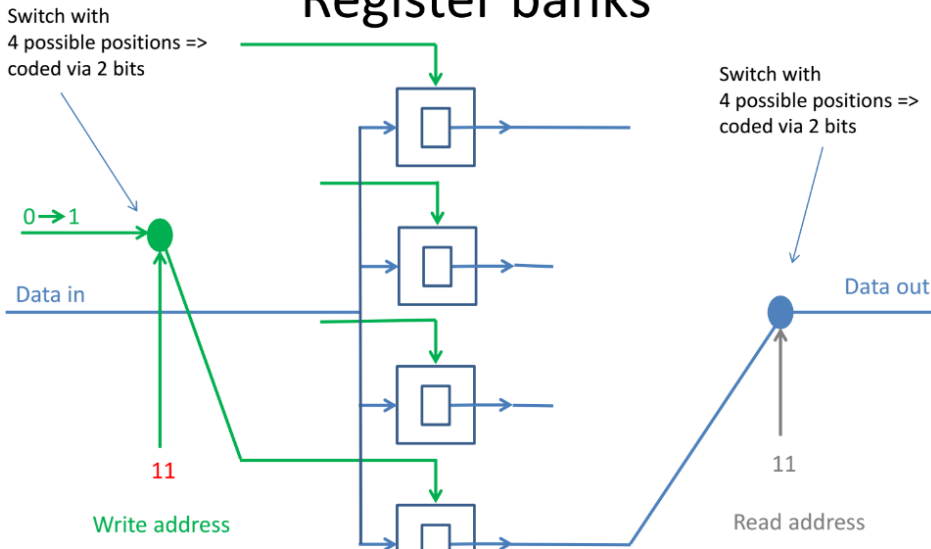
Reasoning forward: "Data out" goes to
 - Another memory element
 - "outside" the processor (I/O)

In both cases, data can pass via combinatorial circuits, e.g. an ALU



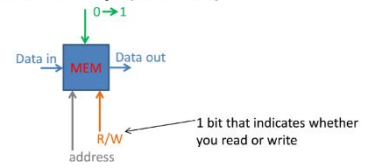
Storage of 8 bits

Register banks

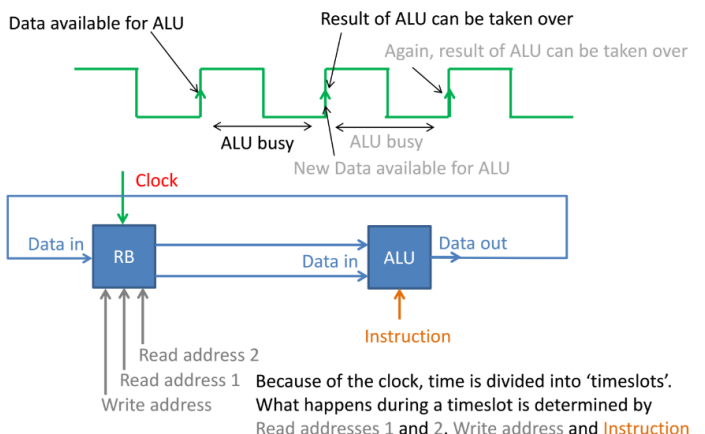


CLOCK SIGNAL:

Memory (MEM)



- Same functionality as Registerbank
- Read and Write not simultaneously => 1 address input + additional line that indicates read or write
- Consists of a lot more memory elements (larger address range)



Because of the clock, time is divided into 'timeslots'.
 What happens during a timeslot is determined by
 Read addresses 1 and 2, Write address and Instruction