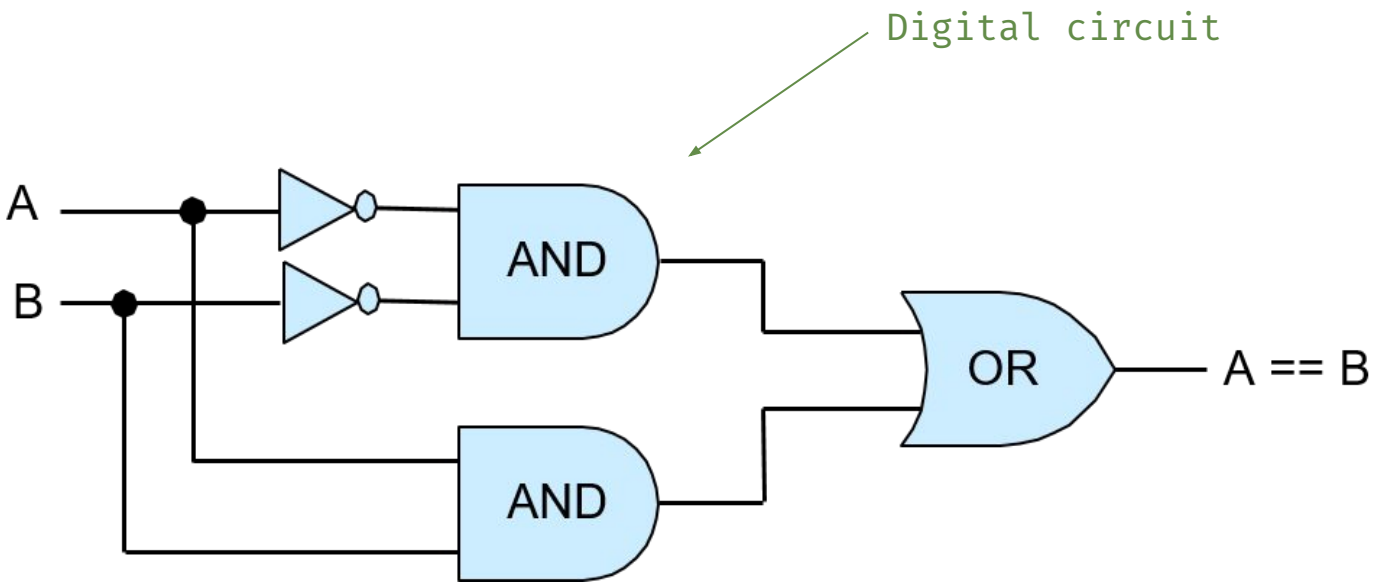
0X BARE C8 METAL

CMPSC 0XC8 ALLEGHENY COLLEGE 0X7E7

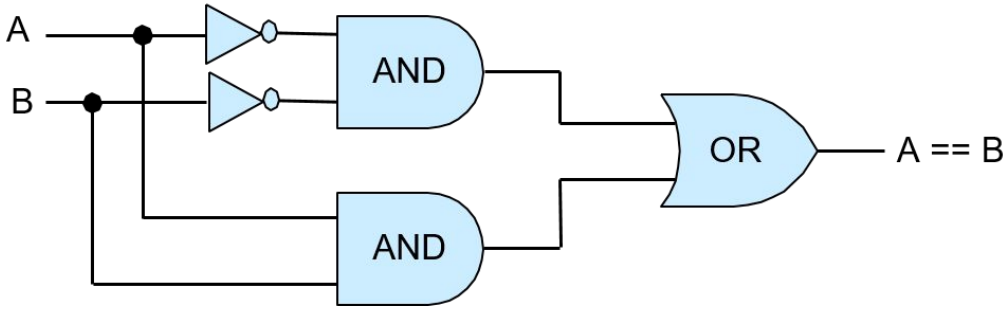


(Checks if two signals (bytes) are equal)

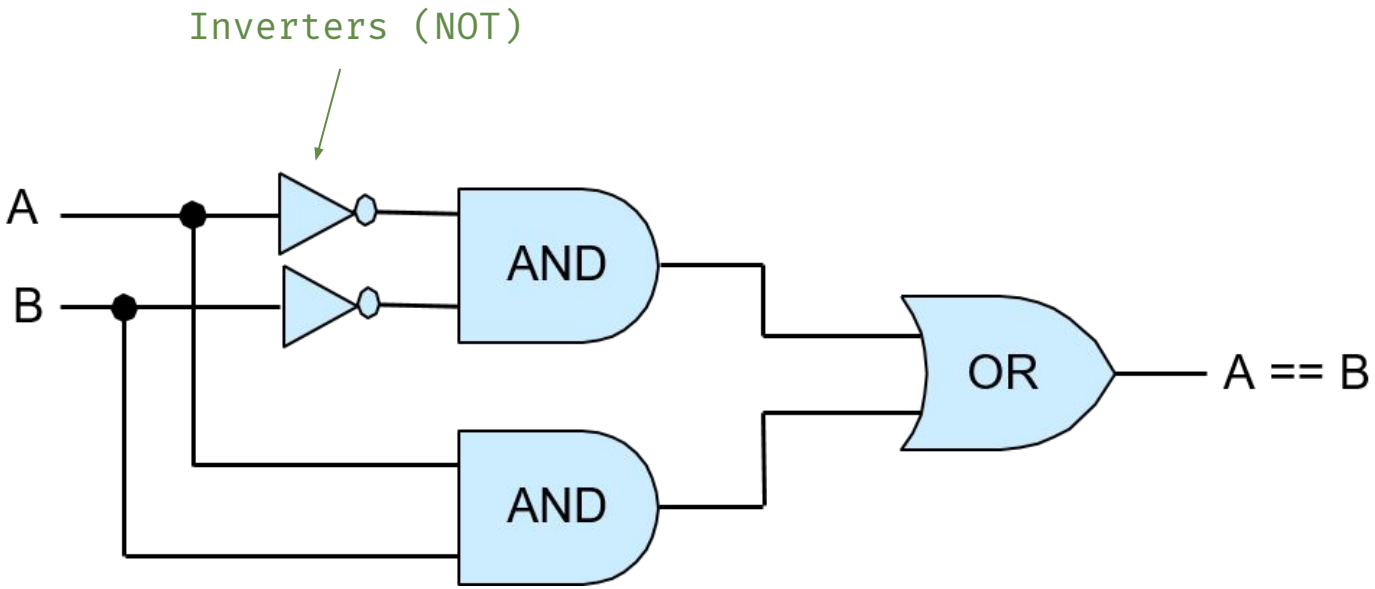
To Tell the Truth

We describe digital circuits through their **outcomes** using a tool called a “truth table.”

A	B	OUTPUT
0	0	1
1	0	0
0	1	0
1	1	1



A	B	OUTPUT
0	0	1
1	0	0
0	1	0
1	1	1



(**NOT A** *AND* **NOT B**) *OR* (**A** *AND* **B**)

Boo!(lean) Algebra

Based in 3 operations:

- AND
- OR
- NOT

(Really, there are more; that would just make things confusing right now)

They express boolean outcomes, but *not exactly* **true**/**false**

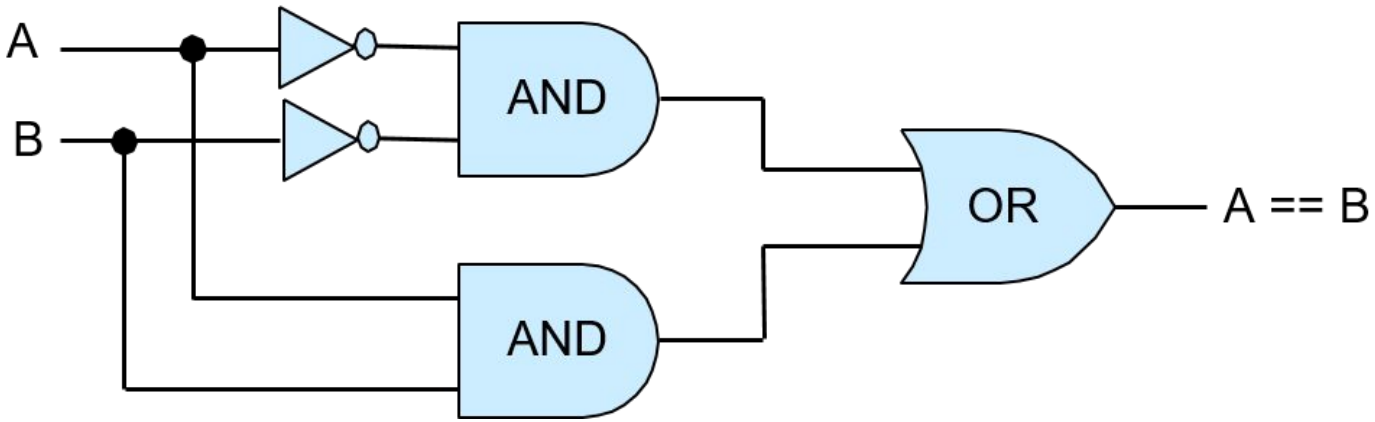
"truthiness" is subjective, based on desired outcome.

Boo!(lean) Algebra

Expression	Reading	Example	Outcome
A ∧ B	A AND B	0 AND 0	0
		0 AND 1	0
		1 AND 0	0
		1 AND 1	1
A ∨ B	A OR B	0 OR 0	0
		0 OR 1	1
		1 OR 0	1
		1 OR 1	1

Boo!(lean) Algebra

Expression	Reading	Example	Outcome
¬A	NOT A	0	1
		1	0



(¬A ∧ ¬B) ∨ (A ∧ B)

(**NOT A** *AND* **NOT B**) *OR* (**A** *AND* **B**)

A	B	OUTPUT
0	0	1

A	B	OUTPUT
0	0	1
1	0	0

A	B	OUTPUT
0	0	1
1	0	0
0	1	0

A	B	OUTPUT
0	0	1
1	0	0
0	1	0
1	1	1

Boolean to Assembly

English	Algebraic	Assembly
AND	$\wedge$	AND
OR	$\vee$	ORR
NOT	$\neg$	NOT*

* Not exactly for ARMv6; some ISAs have it, but you have one provided to you, so you can use NOT

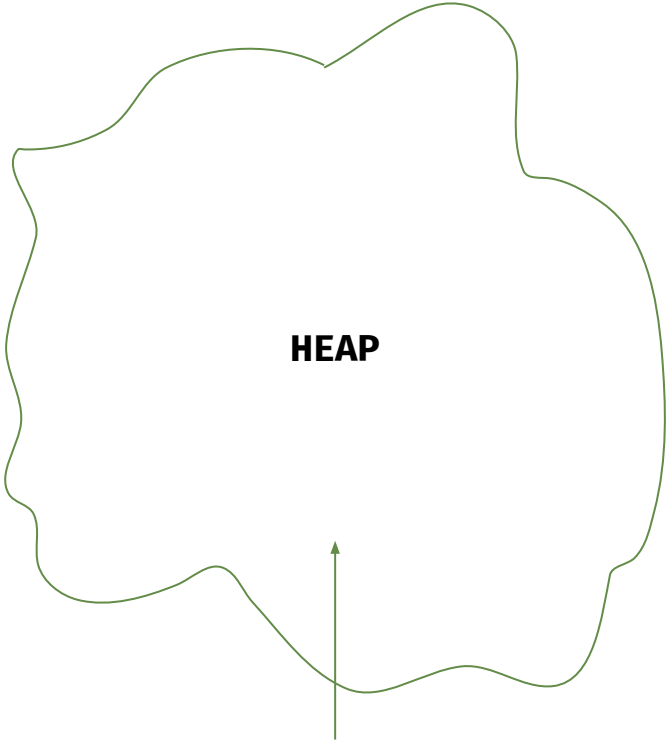
Boolean to Assembly

Assembly	Usage
AND	AND R0, R1
ORR	ORR R0, R1
NOT	NOT R0

Stack and Heap

STACK
0xdef45ea5
0x134a48fd
0x6785efae

Lawful good

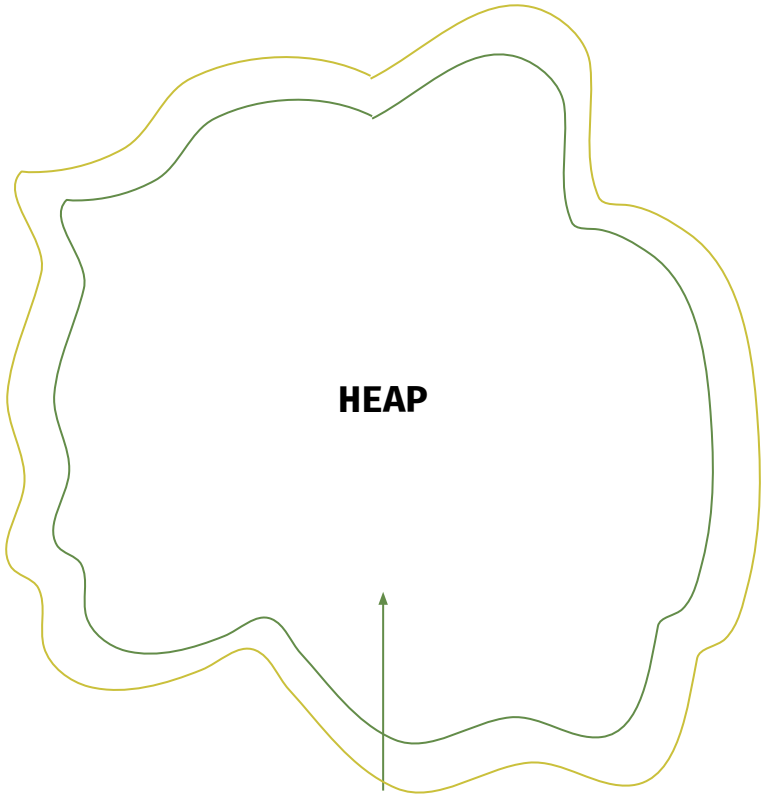


Chaotic neutral

Stack and Heap

STACK
0xdef45ea5
0x134a48fd
0x6785efae

Lawful good



Chaotic neutral

Brief introduction to the “stack”

There exists another quasi-temporary spot for memory which can get us out of a tricky jam: the *stack*.

LIFO
Last In, First Out

VALUE
0xdef45ea5
0x134a48fd
0x6785efae

“grows down”

Brief introduction to the “stack”

PUSH
Places a value in the stack *in order*

REGISTERS	
R0	0xdef45ea5
R1	0x134a48fd
R2	0x6785efae

PUSH {R0-R2}

STACK
0xdef45ea5
0x134a48fd
0x6785efae

Brief introduction to the “stack”

POP
Extracts values from
the stack *in order*

REGISTERS	
R0	0x6785efae
R1	
R2	

POP {R0}

STACK
0xdef45ea5
0x134a48fd
0x6785efae

Brief introduction to the “stack”

POP
Extracts values from
the stack *in order*

REGISTERS	
R0	0x6785efae
R1	0x134a48fd
R2	0xdef45ea5

POP {R1,R2}

STACK
0xdef45ea5
0x134a48fd
0x6785efae

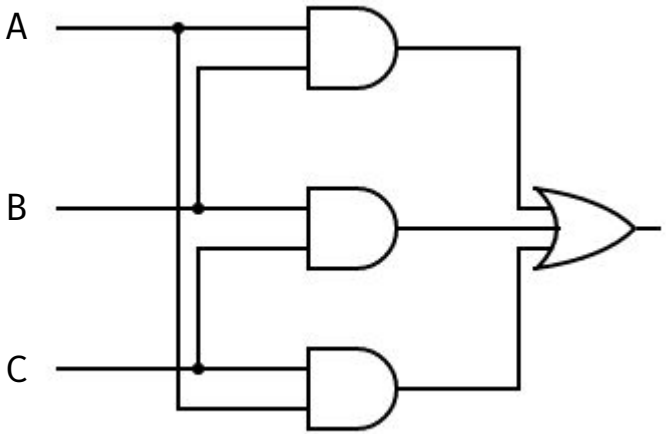
Brief introduction to the “stack”

PUSH $\{R_M, R_D, \ldots\}$ POP $\{R_M, R_D, \ldots\}$

PUSH $\{R_M - R_D\}$ POP $\{R_M - R_D\}$

Return of the Math Test

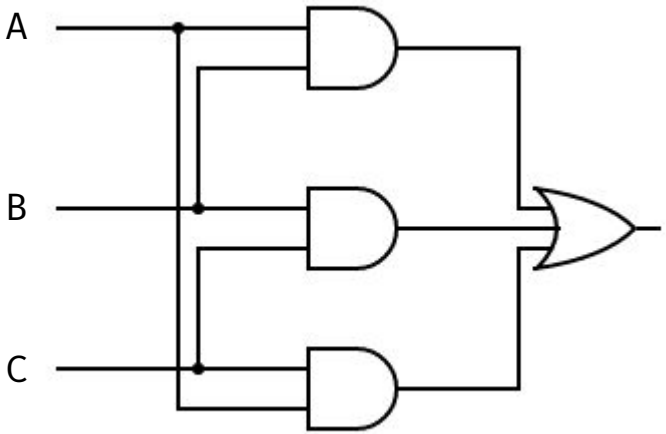
$$(A \wedge B) \vee (A \wedge C) \vee (A \wedge C)$$



A	B	C	OUTPUT
0	0	0	0
0	0	1	0
0	1	0	0
1	0	0	0
0	1	1	1
1	1	0	1
1	1	1	1

Return of the Math Test

$$(A \wedge B) \vee (A \wedge C) \vee (A \wedge C)$$



A	B	C	OUTPUT
0	0	0	0
0	0	1	0
0	1	0	0
1	0	0	0
0	1	1	1
1	1	0	1
1	1	1	1