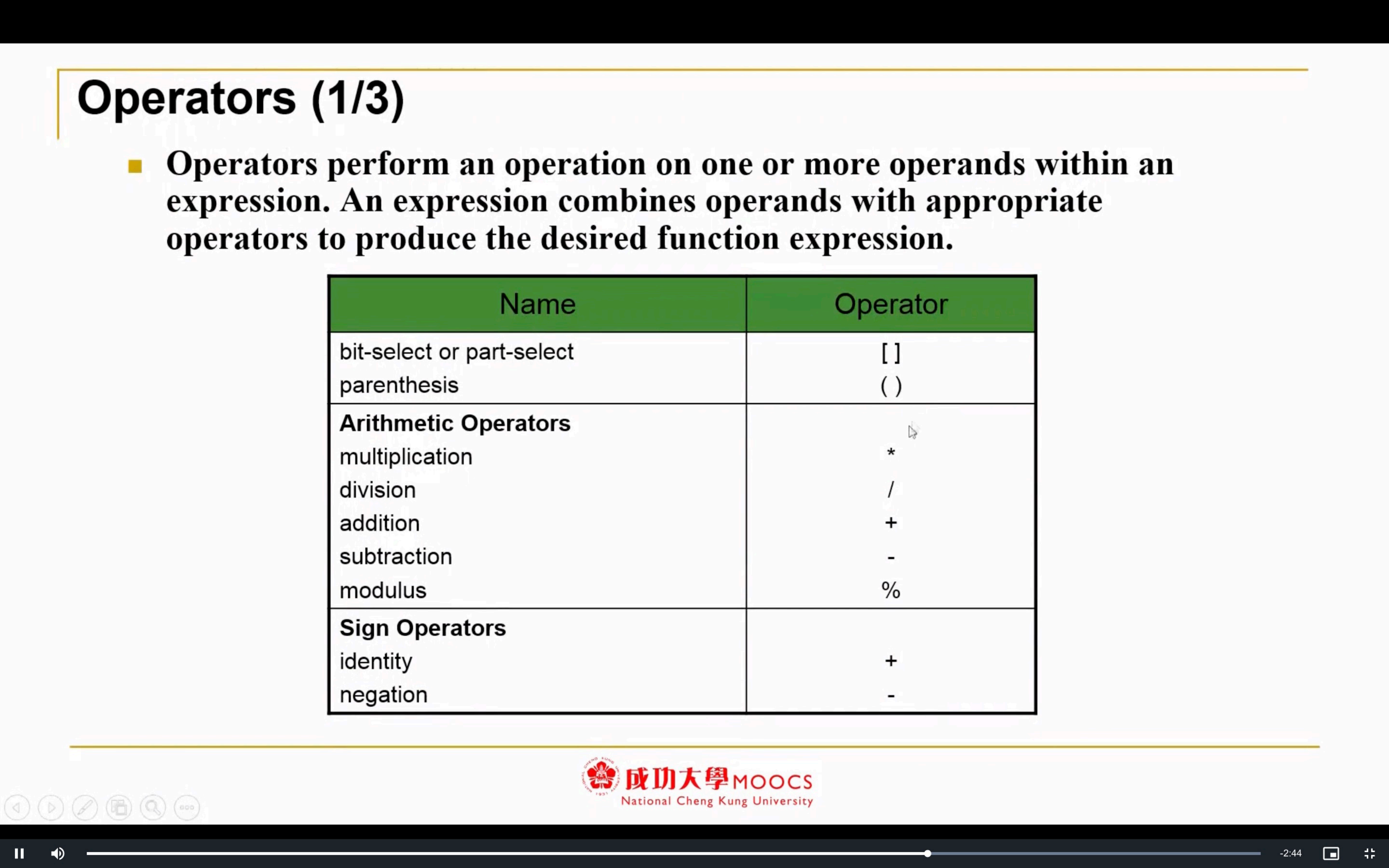
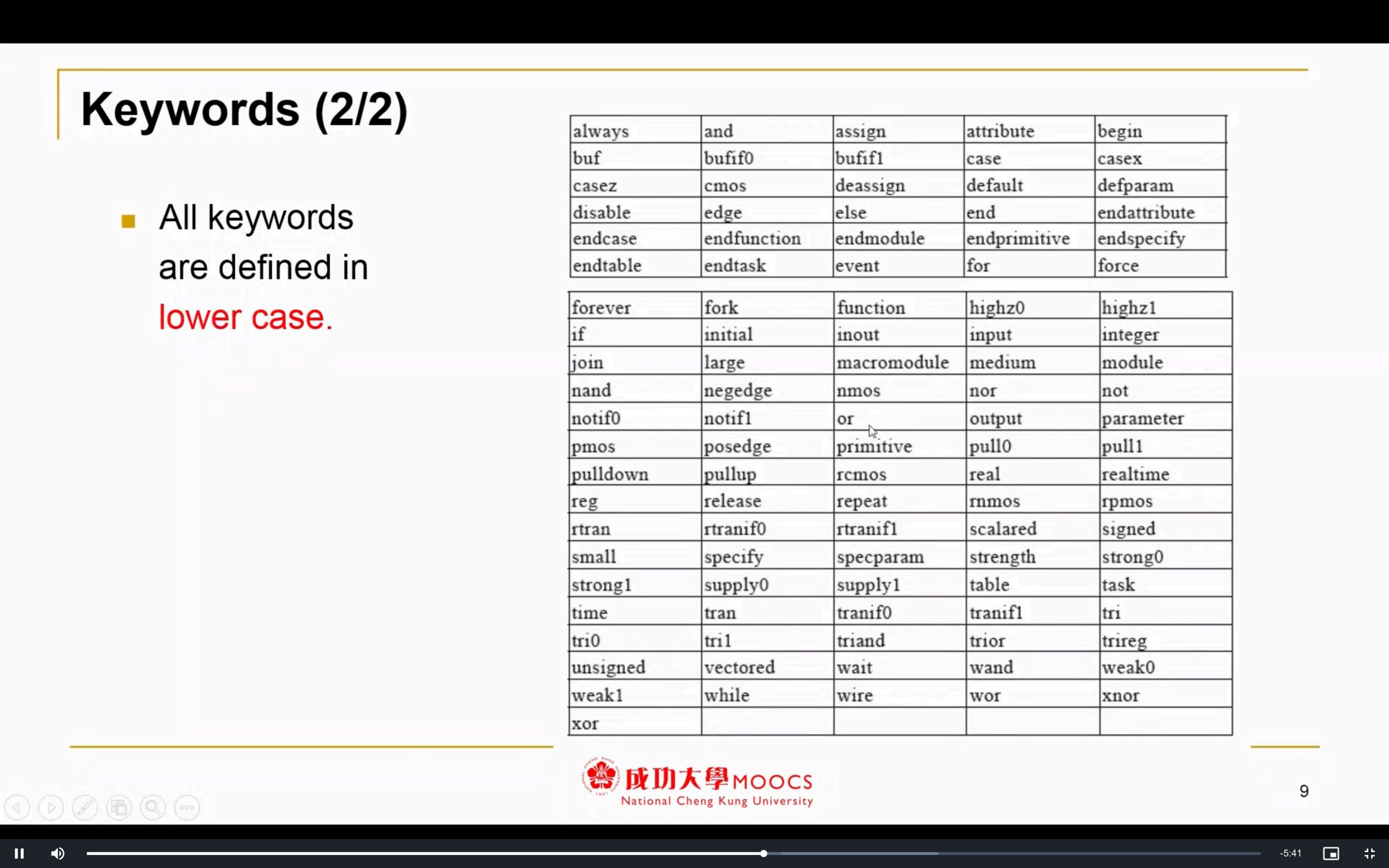


Operators (1/3)

- Operators perform an operation on one or more operands within an expression. An expression combines operands with appropriate operators to produce the desired function expression.

Name	Operator
bit-select or part-select	[ ]
parenthesis	( )
Arithmetic Operators	
multiplication	*
division	/
addition	+
subtraction	-
modulus	%
Sign Operators	
identity	+
negation	-



Keywords (2/2)

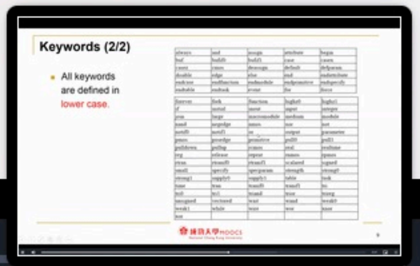
- All keywords are defined in lower case.

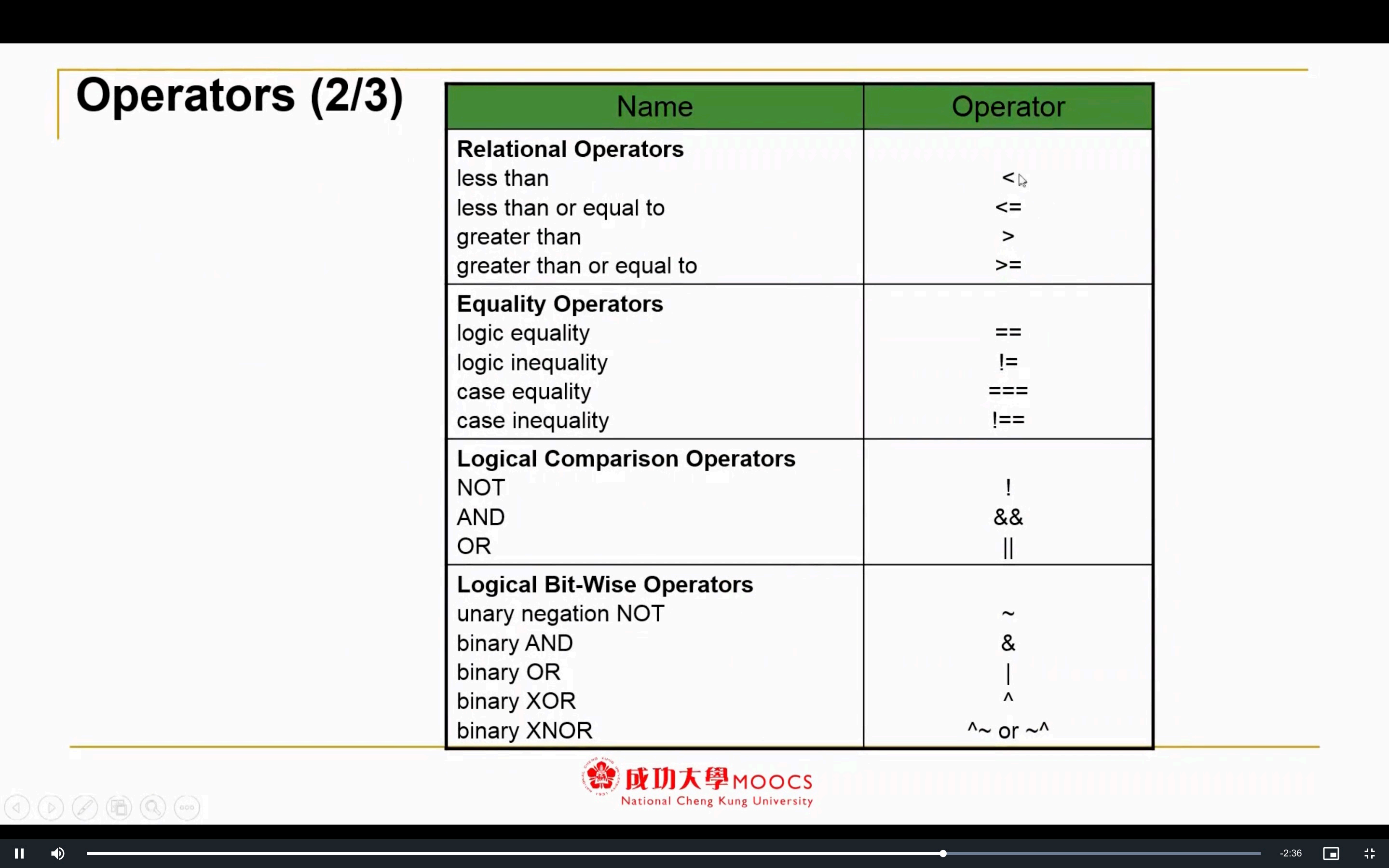
always	and	assign	attribute	begin
buf	bufif0	bufif1	case	casex
casez	cmos	deassign	default	defparam
disable	edge	else	end	endattribute
endcase	endfunction	endmodule	endprimitive	endspecify
endtable	endtask	event	for	force

forever	fork	function	highz0	highz1
if	initial	inout	input	integer
join	large	macromodule	medium	module
nand	negedge	nmos	nor	not
notif0	notif1	or	output	parameter
pmos	posedge	primitive	pull0	pull1
pulldown	pullup	rcmos	real	realtime
reg	release	repeat	rnmos	rpmos
rtran	rtranif0	rtranif1	scalared	signed
small	specify	specparam	strength	strong0
strong1	supply0	supply1	table	task
time	tran	tranif0	tranif1	tri
tri0	tri1	triand	trior	trireg
unsigned	vectored	wait	wand	weak0
weak1	while	wire	wor	xnor
xor				

Data type - net (wire)

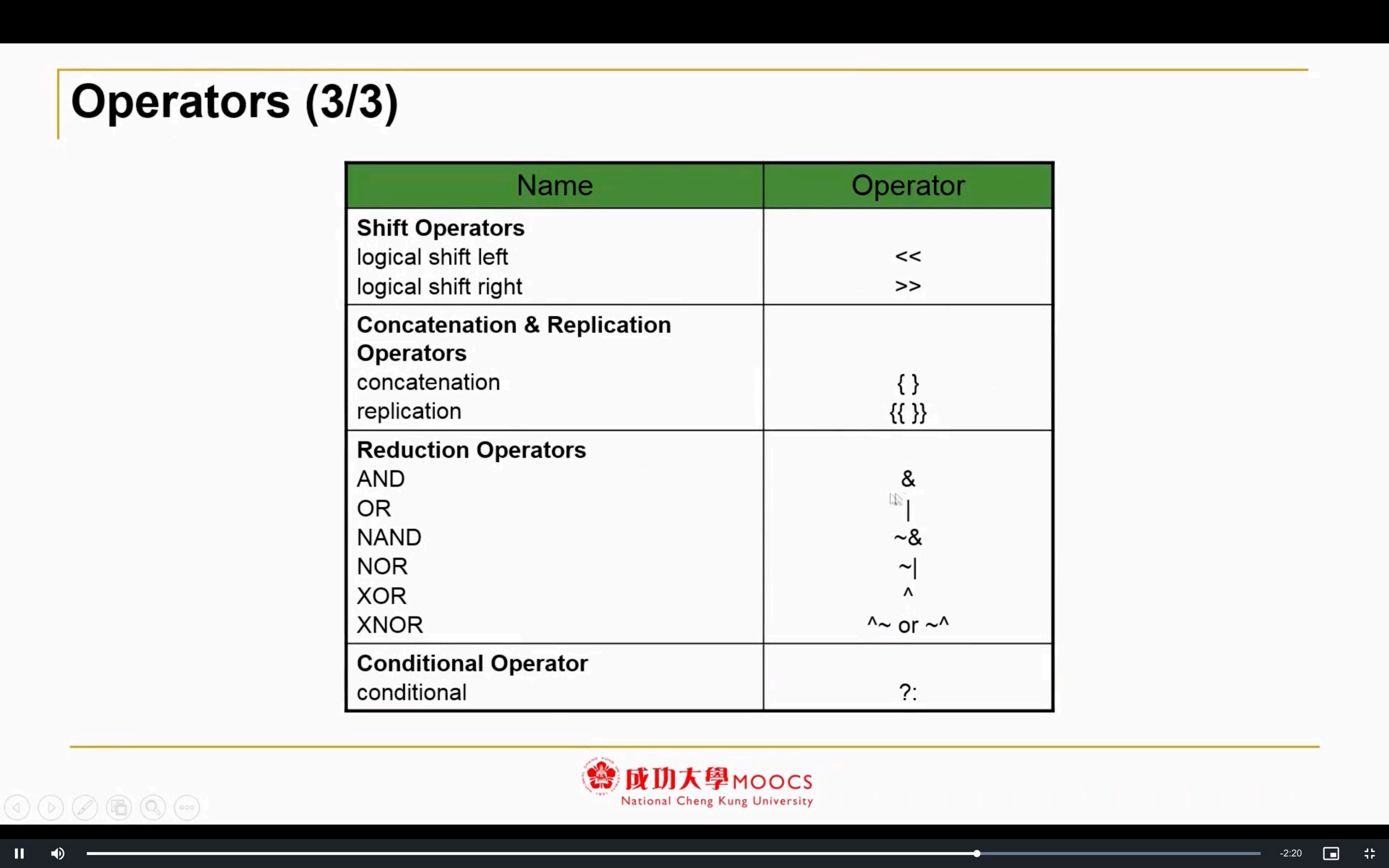
接線種類	功能	能否合成
wire, tri	For standard interconnection wires (default)	可
wor, trior	For multiple drivers that are wire-ORed	wor 可，trior 否
wand, triand	For multiple drivers that are wire-ANDed	wand 可，triand 否
trireg	For nets with capacitive storage	否
tri0	For nets which pull down when not driven	否
tri1	For nets which pull up when not driven	否
supply0	For power rails	可
supply1	For ground rails	可





Operators (2/3)

<table>
<tr><th>Name</th><th>Operator</th></tr>
<tr><td><b>Relational Operators</b></td><td></td></tr>
<tr><td>less than</td><td><</td></tr>
<tr><td>less than or equal to</td><td><=</td></tr>
<tr><td>greater than</td><td>></td></tr>
<tr><td>greater than or equal to</td><td>>=</td></tr>
<tr><td><b>Equality Operators</b></td><td></td></tr>
<tr><td>logic equality</td><td>==</td></tr>
<tr><td>logic inequality</td><td>!=</td></tr>
<tr><td>case equality</td><td>===</td></tr>
<tr><td>case inequality</td><td>!==</td></tr>
<tr><td><b>Logical Comparison Operators</b></td><td></td></tr>
<tr><td>NOT</td><td>!</td></tr>
<tr><td>AND</td><td>&&</td></tr>
<tr><td>OR</td><td>||</td></tr>
<tr><td><b>Logical Bit-Wise Operators</b></td><td></td></tr>
<tr><td>unary negation NOT</td><td>~</td></tr>
<tr><td>binary AND</td><td>&</td></tr>
<tr><td>binary OR</td><td>|</td></tr>
<tr><td>binary XOR</td><td>^</td></tr>
<tr><td>binary XNOR</td><td>^~ or ~^</td></tr>
</table>



Operators (3/3)

Name	Operator
Shift Operators	
logical shift left	<<
logical shift right	>>
Concatenation & Replication Operators	
concatenation	{ }
replication	{{ }}
Reduction Operators	
AND	&
OR	|
NAND	~&
NOR	~|
XOR	^
XNOR	^~ or ~^
Conditional Operator	
conditional	?: