

CSE331-HW2-RAPOR

Module(and32): 32 bitlik iki sayının her bir bitini ,halihazırda bulunan “and” modülleri ile işleme sokup sonuç 32 bitlik sonuç sayısının her bir biti ile ifade edilir.

Module(xor_32): 32 bitlik iki sayının her bir bitini ,halihazırda bulunan “xor” modülleri ile işleme sokup sonuç 32 bitlik sonuç sayısının her bir biti ile ifade edilir.

Module(or_32): 32 bitlik iki sayının her bir bitini ,halihazırda bulunan “or” modülleri ile işleme sokup sonucu 32 bitlik sonuç sayısının her bir biti ile ifade edilir.

Module(nor_32): 32 bitlik iki sayının her bir bitini ,halihazırda bulunan “nor” modülleri ile işleme sokup sonucu 32 bitlik sonuç sayısının her bir biti ile ifade edilir.

Module(shifter): 5 katmandan oluşan bir modüldür.Çünkü elimizdeki veriyi en fazla 31 bit kaydırabiliriz ve 31 sayısında en az 5 bit ile ifade edilebilir.İlk katman 2^0 yani 1 basamak kaydırma ,ikinci katman 2^1 yani 2 basamak,üçüncü katman 2^2 yani 4 basamak kaydırma,dördüncü basamak 2^3 basamak kaydırma,beşinci katman 2^4 yani 16 basamak kaydırma yapmak için kullanılır.Eğerki 5 bitinde hepsi 1 değeri ile kurulursa elimizdeki veri 31 bit sağa kaymış olur.(right_shift).Her bir katmanda 32 adet 2:1 lik muxlar vardır.ilik katman muxlarının bir girişi verinin karşılık gelen basamağı ,diğer girişi ise karşılık gelen basamağın bir basamak soludur.en sondaki muxin 1 girişi verimizin işaret biti yani most significant biti ile değerlendirir.Bu şekilde devam ederek her basamaktaki muxların 1 girişleri sırasıyla kendilerinin 2,4,8,16 basamak solundaki bitlerle değerlendirir.açıkta kalan muxlar ise yine işaret bitiyle bağlanır.

Module(left_shifter): shifter(right_shifter) modülünden farklı olarak her mux'un 1 girişi sırasıyla kendilerinin 1,2,4,8,16 basamak sağındaki bitler ile değerlendirir.Ayrıca yine shifter(right_shifter) modülünden farklı olarak 1 girişleri açıkta kalan mux'ların bu girişleri 0 ile değerlendirir.

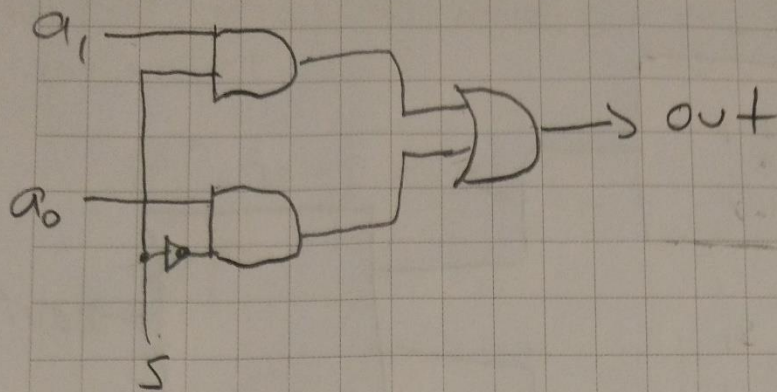
Module(full_adder_32): İçinde toplam 32 adet full_adder modülü çağırılır yani kullanılır.İlk full_adder cin inputu olarak 0 bitini alacak şekilde kurgulanmıştır.ilik full_adder carry_out değeri gelecek full_adder'in cin girişi olarak değerlendirir.Her adder'in 1 bitlik çıkışları ise sum [31:0] outputun her bir bitine atanır. Bu işlem 32 sefer sırasıyla devam eder.

Module(full_minus_31): Aslında iç yapı olarak full_adder_32 modülü ile neredeyse aynıdır.Tek fark çıkarılacak olan 32 bitlik sayının yerine (aslında çıkarmada bir toplama işlemidir) kendisiyle “xor” işlemine sokulmasıyla elde edilen sayının kullanılması ve cin olarak 1 bit değeri uygulanmasıyla elde edilir.

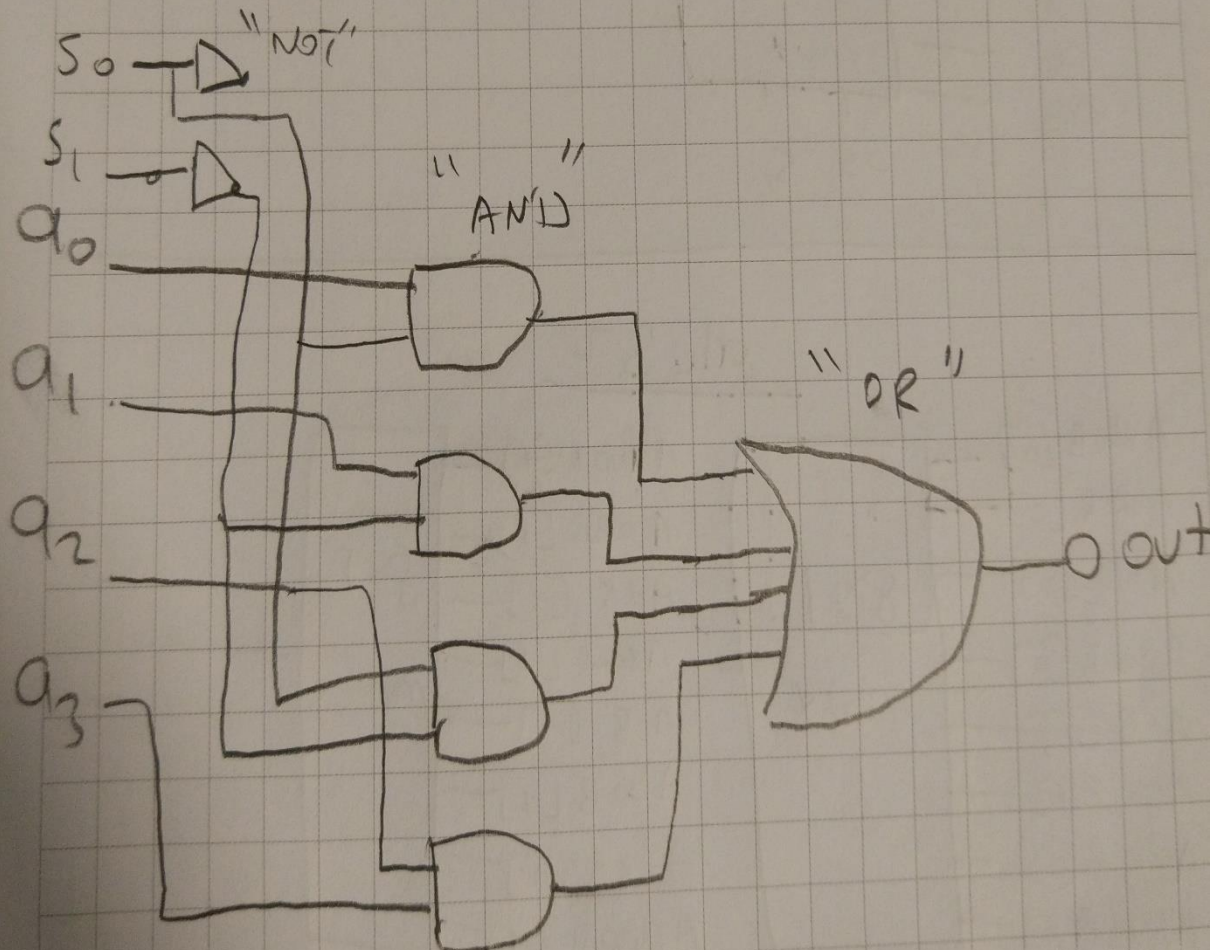
Schematic designs

-Resimlerde gözükmese de 32 bitlik and, or xor ,nor operasyonları tanımlanmış olup ALU içinde kullanılmıştır.İçlerinde 32 tane her biti işleme sokan and ,or gibi operasyonlar vardır.

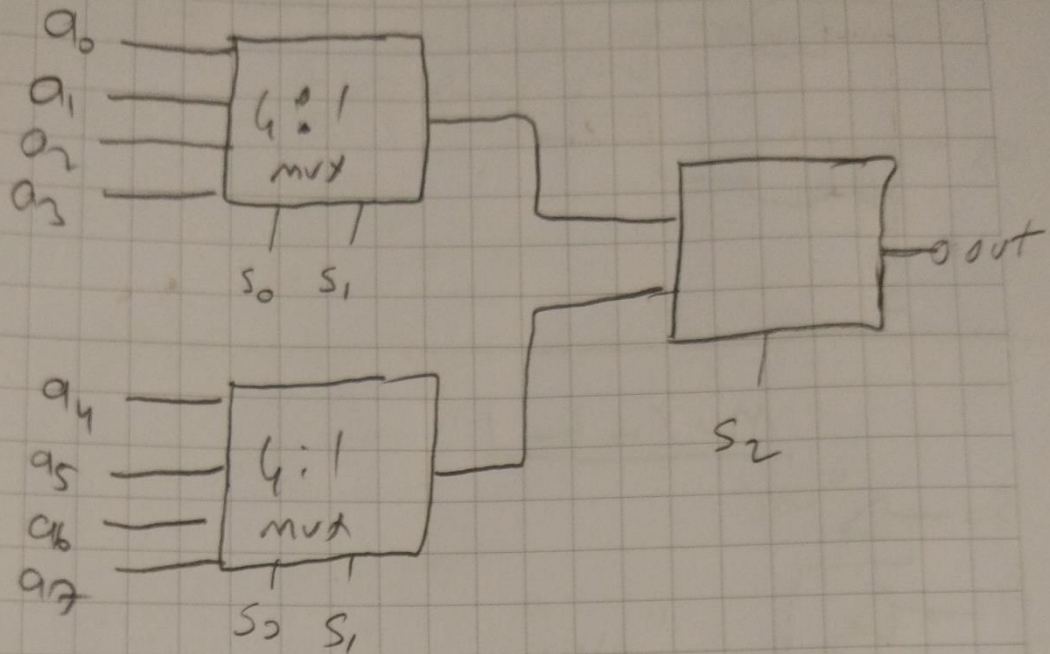
MUX-2-1



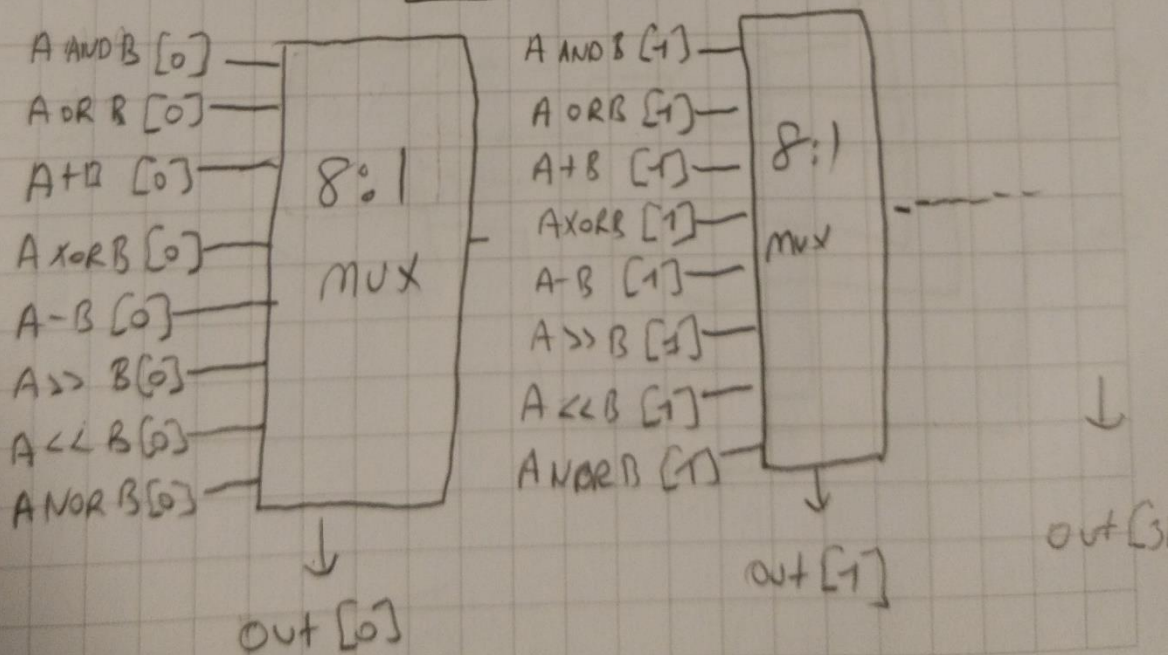
MUX-4-1



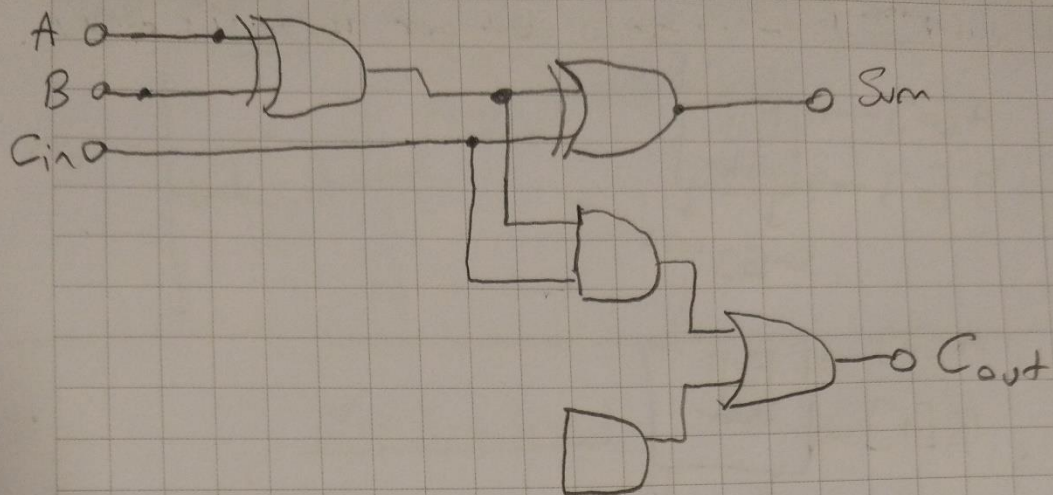
mux_8-1



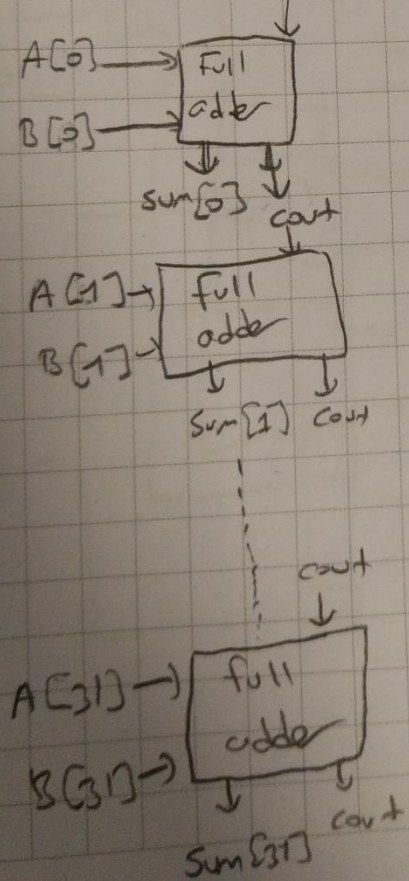
alu32



full-adder



Full-adder-32



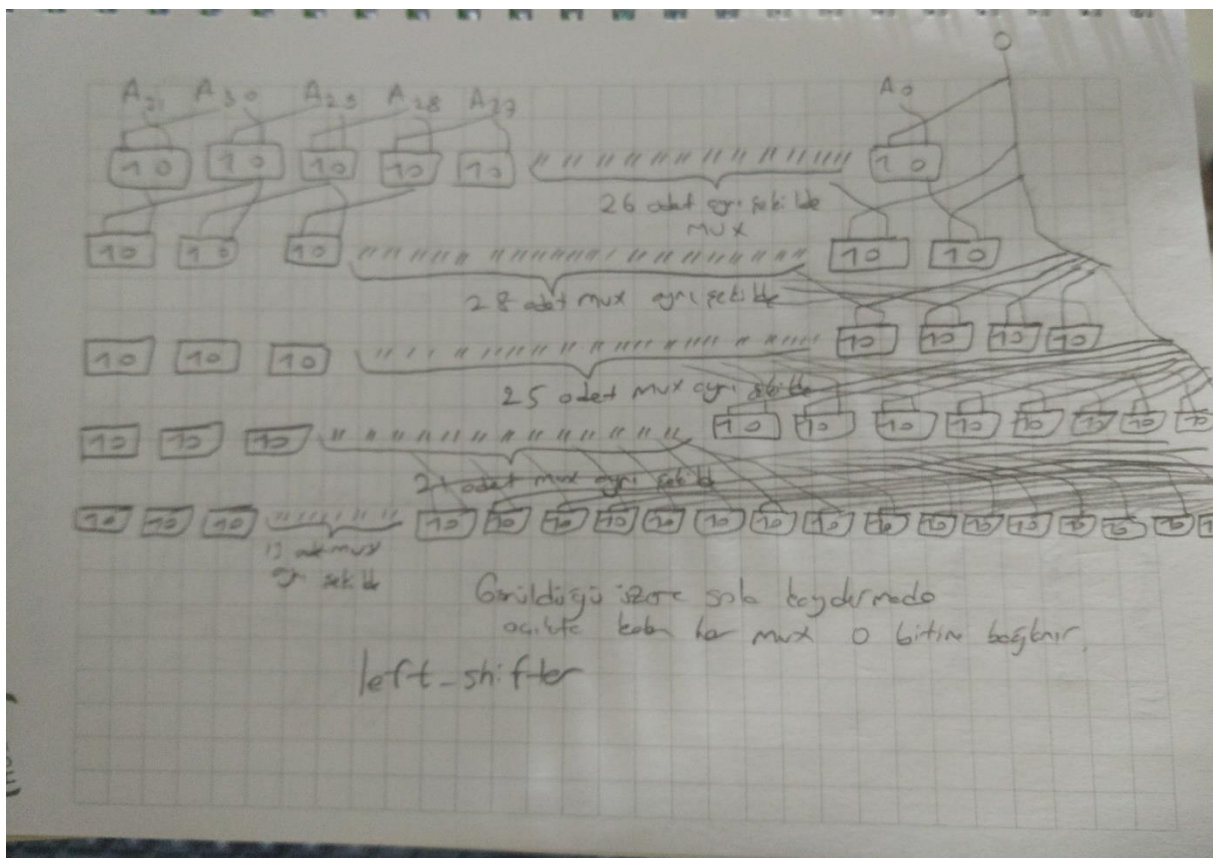
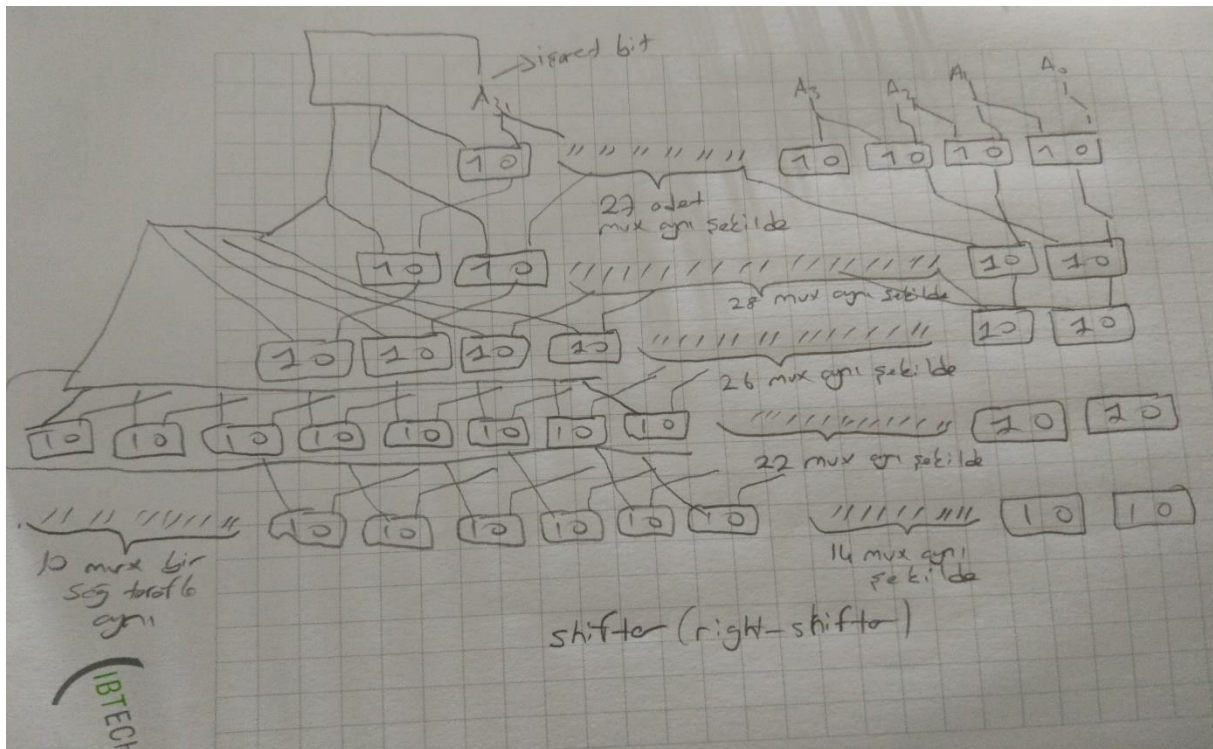
full-mixers-32

B → 32b'11111111 → XOR → B' cin (0 above get isode 1'e dörer)

A[0] → Full
adder
B'[0] → Full
adder
↓ sum[0] cout

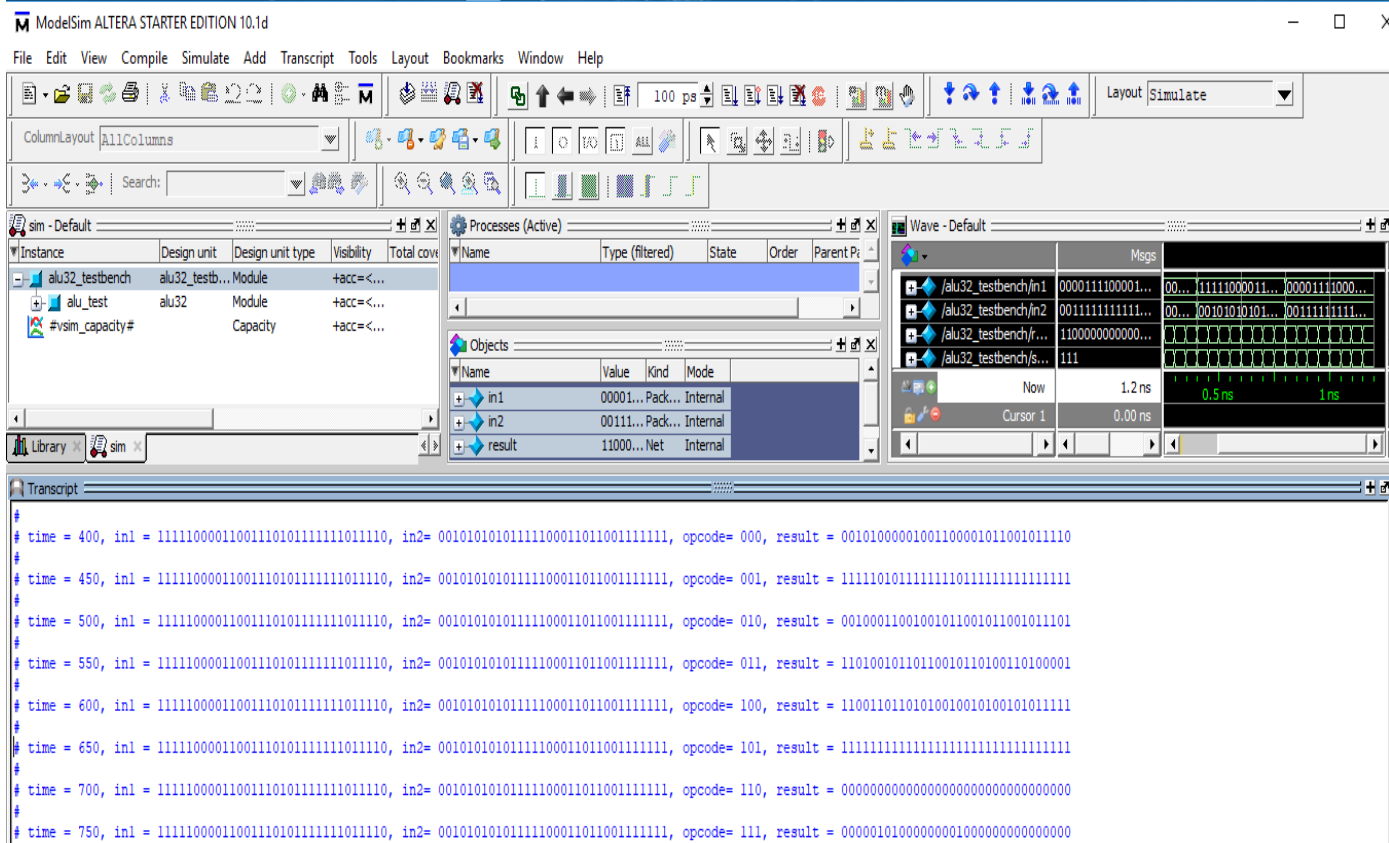
A[1] → Full
adder
B'[1] → Full
adder
↓ sum[1] cout

⋮
A[31] → Full
adder
B'[31] → Full
adder
↓ sum[31] cout



Simulations results:

Case_1:



Case3:

ModelSim ALTERA STARTER EDITION 10.1d

File Edit View Compile Simulate Add Transcript Tools Layout Bookmarks Window Help

ColumnLayout AllColumns

sim - Default

Instance	Design unit	Design unit type	Visibility	Total cov
alu32_testbench	alu32_testb...	Module	+acc=<...	
alu_test	alu32	Module	+acc=<...	
#vsim_capacity#		Capacity	+acc=<...	

Processes (Active)

Name	Type (filtered)	State	Order	Parent P

Objects

Name	Value	Kind	Mode
in1	00001... Pack...	Internal	
in2	00111... Pack...	Internal	
result	11000... Net	Internal	

Wave - Default

Signal	Value	Kind	Mode
/alu32_testbench/in1	0000111100001...		
/alu32_testbench/in2	0011111111111...		
/alu32_testbench/r...	1100000000000...		
/alu32_testbench/s...	111		

Transcript

```
# time = 800, in1 = 00001111000011110011111100011110, in2= 0011111111111100011011001100001, opcode= 000, result = 00001111000011100011011000000000
# time = 850, in1 = 00001111000011110011111100011110, in2= 0011111111111100011011001100001, opcode= 001, result = 00111111111111110011111101111111
# time = 900, in1 = 00001111000011110011111100011110, in2= 0011111111111100011011001100001, opcode= 010, result = 01001111000011010111010101111111
# time = 950, in1 = 00001111000011110011111100011110, in2= 0011111111111100011011001100001, opcode= 011, result = 00110000111100010000100101111111
# time = 1000, in1 = 00001111000011110011111100011110, in2= 0011111111111100011011001100001, opcode= 100, result = 11001111000100010000100010111101
# time = 1050, in1 = 00001111000011110011111100011110, in2= 0011111111111100011011001100001, opcode= 101, result = 00000111100001111001111100011110
# time = 1100, in1 = 00001111000011110011111100011110, in2= 0011111111111100011011001100001, opcode= 110, result = 0001111000011110011111000111100
# time = 1150, in1 = 00001111000011110011111100011110, in2= 0011111111111100011011001100001, opcode= 111, result = 11000000000000011000000100000000
#
VSI6> add wave -position insertpoint \
sim:/alu32_testbench/in1 \
sim:/alu32_testbench/in2 \
sim:/alu32_testbench/result \
sim:/alu32_testbench/select
```

Windows'u Etkinleştir
Windows'u etkinleştirmek için Ayarlar'a gidin.