Θέματα 3ης άσκησης

1. Σκοπός της άσκησης

Στην άσκηση αυτή συνεχίζουμε στο online Chisel bootcamp. Επισκεφθείτε τη διεύθυνση:

https://mybinder.org/v2/gh/freechipsproject/chisel-bootcamp/master

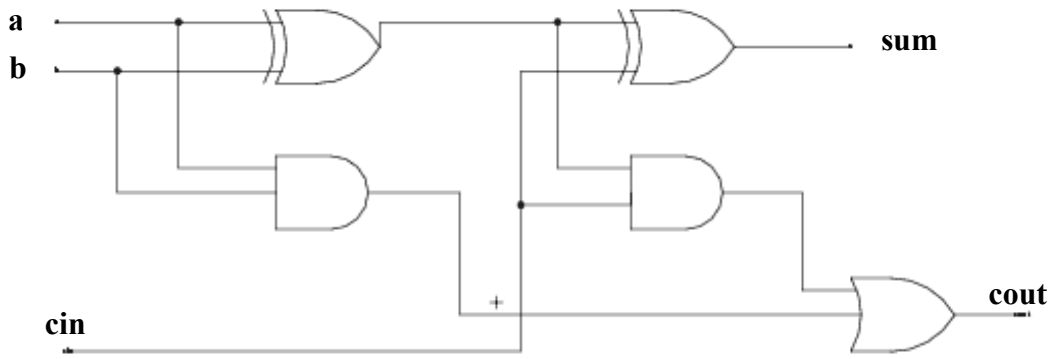
Σκοπός της άσκησης αυτής είναι η κατασκευή ενός πλήρους αθροιστή (full adder) και η χρήση του σε ένα μεγαλύτερο module. Χρησιμοποιήστε το notebook της 3ης άσκησης που θα βρείτε στο site του μαθήματος, ανεβάζοντάς το στο online Chisel bootcamp. Στη συνέχεια συμπληρώστε τα κελιά σύμφωνα με τις οδηγίες.

2. Ο πλήρης αθροιστής.

Ο πλήρης αθροιστής προσθέτει 2 bits εισοδου (**a** και **b**) και ένα κρατούμενο εισόδου (**cin**) και παράγει ως εξόδους ένα άθροισμα (**sum**) και ένα κρατούμενο εξόδου (**cout**) σύμφωνα με τον παρακάτω πίνακα αλήθειας:

a	b	cin	sum	cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Υπάρχουν πολλοί τρόποι για να σχεδιάσετε έναν πλήρη αθροιστή. Ένας απλός τρόπος φαίνεται στο πιο κάτω σχήμα:



Χρησιμοποιώντας το αντίστοιχο κελί του notebook (θα το βρείτε με τις εισόδους/εξόδους προσυμπληρωμένες) προσθέστε τη λογική για την υλοποίηση του προηγούμενου σχήματος.

Υπόδειξη: Μπορείτε να χρησιμοποιήσετε προσωρινές μεταβλητές για το κύκλωμά σας, π.χ.:

```
val temp = io.a ^ io.b
```

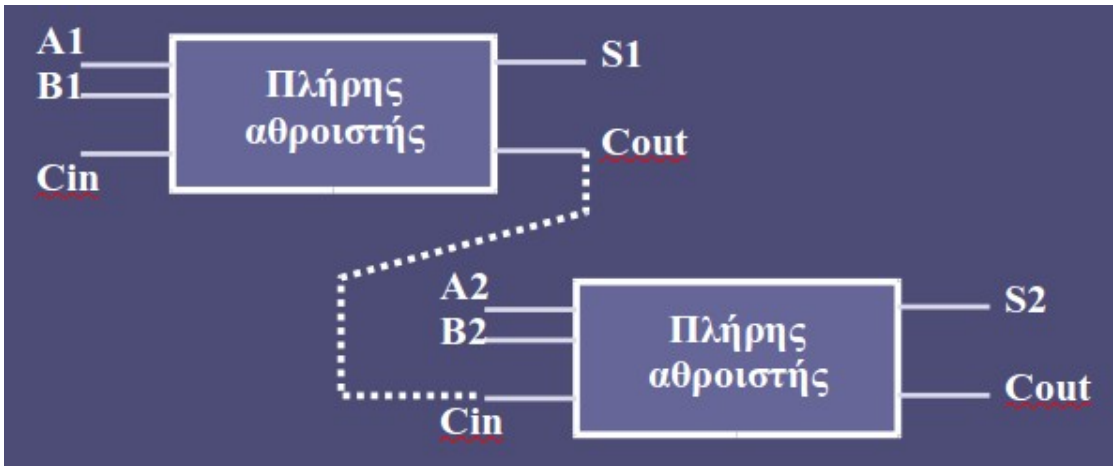
και να τις χρησιμοποιήσετε αργότερα π.χ.:

```
io.sum := sum1 ^ io.cin
```

Βεβαιωθείτε ότι το κύκλωμά σας λειτουργεί όπως πρέπει για όλους τους συνδυασμούς εισόδων σύμφωνα με τον πίνακα αλήθειας του πλήρους αθροιστή. Συμπληρώστε τον κώδικα ελέγχου στου επόμενο κελί του notebook.

3. Αθροιστής 4 bits.

Πολλοί πλήρεις αθροιστές μπορούν να συνδυαστούν για να προσθέτουμε εισόδους με εύρος μεγαλύτερο από 1 bit. Ο τρόπος διασύνδεσης (ripple carry adder) φαίνεται στο επόμενο σχήμα και μπορεί να επεκταθεί και για μεγαλύτερο αριθμό bits:



Κατασκευάστε ένα αθροιστή **Adder4** όπου τα **a**, **b** και **sum** έχουν εύρος 4 bits, συμπληρώνοντας τον κώδικα που λείπει στο αντίστοιχο κελί.

Παρατηρήστε πώς μπορούμε να χρησιμοποιήσουμε ένα module που έχουμε ήδη φτιάξει, π.χ.:

```
val adder0 = Module(new FullAdder())
```

Στη συνέχεια δοκιμάστε το κύκλωμα που προκύπτει με διάφορες ενδεικτικές τιμές.

4. Τελειώνοντας το εργαστήριο

Μπορείτε να δείτε στο τελευταίο κελί του notebook έναν τρόπο για να δοκιμάσετε τον Adder4 με τυχαίες τιμές.

Όταν ολοκληρώσετε την άσκηση, κατεβάστε (**download**) το notebook με τον κώδικά σας. **Φυλάξτε το για τα επόμενα εργαστήρια!**