Πανεπιστήμιο Κρήτης
Τμήμα Επιστήμης Υπολογιστών
ΗΥ-225: Οργάνωση Υπολογιστών
Δ. Πνευματικάτος

ΗΥ-225
Άνοιξη 2000
1999-04-5 – βδομάδα 10
φυλλάδιο 11

6η Σειρά Ασκήσεων

Περιγραφή Επεξεργαστή σε Verilog και Προσομοίωση

Σε ομάδες των δύο ατομων

Προθεσμία έως Τετάρτη 10 Μαΐου & Τετάρτη 17 Μαΐου (βδομάδες 12-13)

Περιγράψτε τον επεξεργαστή σας (datapath και έλεγχος) της 4ης σειράς ασκήσεων on-line, σε Verilog. Η περιγραφή του κυκλώματός σας θα είναι "δομική" (structural) και όχι "συμπεριφοράς" (behavioral) και δεν θα λαμβάνει σοβαρά υπ'όψη καθυστερήσεις. Η αρχικοποίηση του κυκλώματός σας θα γίνεται μόνο με φόρτωση της μνήμης (πρόγραμμα, δεδομένα) από αρχείο, και με ενεργοποίηση ενός σήματος εισόδου "reset" μέχρι λίγο μετά την πρώτη ενεργή ακμή του ρολογιού.

Χρησιμοποιήστε δομημένη περιγραφή ώστε να μπορείτε να ελέγξετε τα διάφορα κομμάτια του κυκλώματός σας ανεξάρτητα. Στείλτε την περιγραφή σας αυτή στο λογαριασμό του μαθήματος (hy225) με e-mail ή με το πρόγραμμα submit (λεπτομέρειες θα δωθούν αργότερα) μέχρι τη Παρασκευή 10 Μαΐου.

Στη συνέχεια, ελέγξτε εξονυχιστικά, με μακρύς ελέγχους (test patterns) και διορθώστε πλήρως (debugging) το σχέδιο σας. ΄Εχετε προθεσμία μέχρι τη Τετάρτη 17 Μαΐου. Μετά απ' αυτό θα εξεταστείτε προφορικά στο σχέδιο σας και την προσομοίωσή του (και θα χρειαστεί να αποδείξετε ότι το κάνατε μόνοι σας και δεν το αντιγράψατε...).

Η μονάδα χρόνου που θα χρησιμοποιήστε θα είναι το 1 ns, και αυτό θα το δηλώστε στην αρχή της περιγραφής σας, λέγοντας:

```
'timescale 1 ns / 1 ns
```

(ο πρώτος αριθμός είναι η χρονική μονάδα και ο δεύτερος είναι η ακρίβεια). Γιά την περιγραφή συνδυαστικής λογικής θα χρησιμοποιήστε εντολές "continuous assignment" (βλ. παραγρ. 2.4, 2.5 του μικρού εγχειριδίου που σας δόθηκε). Γιά παράδειγμα μιά πύλη and-or-invert 2+3=5 εισόδων περιβράφεται ως εξής:

```
wire mux3_sel_nxt;
assign mux3_sel_nxt = ~((case1 & case2) | (c3 & c4 & enab5));
```

Σχεδιάστε και περιγράψτε το κύκλωμά σας χρησιμοποιόντας τη βιβλιοθήκη έτοιμων υποσυστημάτων (modules):

```
~hy225/veriwell/lib/ask6_lib.v
```

Η βιβλιοθήκη αυτή περιέχει τα εξής modules:

`RegLd(Q, D, Ld, clk)`
Ακμοπυροδότητος καταχωρητής θετικής ακμής με είσοδο επίτρεψης φόρτωσης. ΄Εχει μία παράμετρο, width, που είναι το πλάτος του (πλήθος των bits του), με τιμή default = 32 bits (κάνοντάς την 1 παίρνουμε ακμοπυροδότητο flip-flop). Πόρτες: Q: έξοδος (πλάτος width), D: είσοδος (πλάτος width), Ld: είσοδος επίτρεψης φόρτωσης (πλάτος 1), clk: ρολόϊ. Ο χρόνος προετοιμασίας (set-up time) είναι 0 ns και ο χρόνος κρατήματος (hold time) είναι 0, τόσο για το D όσο και γιά το Ld. Η καθυστέρηση εξόδου (Q από clk) είναι 2 ns.

`Mux2(Out, In0, In1, S)`
Πολυπλέκτης 2-σε-1. ΄Εχει μια παράμετρο, width, η οποία είναι το πλάτος του (πλήθος data bits), με τιμή default = 32 bits. Πόρτες: Out: έξοδος (πλάτος width), In0: είσοδος επιλεγόμενη όταν S=0 (πλάτος width), In1: είσοδος επιλεγόμενη όταν S=1 (πλάτος width), S: είσοδος επιλογής (πλάτος 1). Η καθυστέρηση εξόδου (Out από In0, In1, S) είναι 2 ns.

`Mux4(Out, In0, In1, In2, In3, S)`

Πολυπλέκτης 4-σε-1. ΄Εχει μια παράμετρο, width, η οποία είναι το πλάτος του (πλήθος data bits), με τιμή default = 32 bits. Πόρτες κατ' αναλογία προς τον Mux2() (η S έχει πλάτος 2). Η καθυστέρηση εξόδου (Out από In*, S) είναι 3 ns.

`Mux8(Out, In0, In1, In2, In3, In4, In5, In6, In7, S)`

Πολυπλέκτης 8-σε-1. ΄Εχει μια παράμετρο, width, η οποία είναι το πλάτος του (πλήθος data bits), με τιμή default = 32 bits. Πόρτες κατ' αναλογία προς τον Mux2() (η S έχει πλάτος 3). Η καθυστέρηση εξόδου (Out από In*, S) είναι 4 ns.

`ALU(ALUout, overflow, zero, A, B, op)`

Αριθμητική-Λογική Μονάδα πλάτους 32 bits. Πόρτες: ALUout: έξοδος δεδομένων (πλάτος width), overflow: έξοδος ένδειξης υπερχείλισης (πλάτος 1), zero: έξοδος πλάτους 1 που ενεργοποιείται όποτε ALUout==0, A: πρώτη είσοδος δεδομένων (πλάτος width), B: δεύτερη είσοδος δεδομένων (πλάτος width), op: είσοδος είδους πράξης πλάτους 3 bits, με έγκυρες τιμές 000 γιά πρόσθεση, 100 γιά αφαίρεση, 001 γιά λογικό ΚΑΙ, 011 γιά λογικό ΄Η. Η καθυστέρηση εξόδου (ALUout από A, B, op) είναι 10 ns.

`RegisterFile(Dout1, Dout2, Din, AR1, AR2, AW, WE, clk)`

Δίπορτο αρχείο καταχωρητών 32 λέξεων επί 32 bits. Πόρτες: Dout1: πρώτη έξοδος δεδομένων (πλάτος 32 bits), Dout2: δεύτερη έξοδος δεδομένων (πλάτος 32 bits), Din: είσοδος δεδομένων (πλάτος 32 bits), AR1: είσοδος διεύθυνσης πρώτης ανάγνωσης (πλάτος 5 bits), AR2: είσοδος διεύθυνσης δεύτερης ανάγνωσης (πλάτος 5 bits), AW: είσοδος διεύθυνσης εγγραφής (πλάτος 5 bits), WE: είσοδος επίτρεψης εγραφής (1 bit), clk: ρολόϊ εγγραφής. Η ανάγνωση είναι ασύγχρονη, και γίνεται όποτε WE==0. Η εγγραφή είναι σύγχρονη, και γίνεται στη θετική ακμή του ρολογιού, όταν WE==1. Ο χρόνος προετοιμασίας (set-up time) των Din, AW, και WE πριν από τη θετική ακμή του ρολογιού είναι 0 ns. Η καθυστέρηση εξόδου (Dout1, Dout2 από AR, AR2) είναι 5 ns.

`Memory(Dout, Din, Address, RE, WE, clk)`

Στατική μνήμη τυχαίας προσπέλασης (SRAM), πλάτους 32 bits, και μεγέθους μέχρι 1 Gwords. Παράμετρος (num_words): το πλήθος των λέξεων που υλοποιεί ο προσομοιωτής (τιμή default = 128). Οι λέξεις που υλοποιεί ο προσομοιωτής είναι από τη διεύθυνση 0 ως και τη διεύθυνση num_words-1. Εγγραφή στο μη υλοποιημένο τμήμα της μνήμης αγνοείται μετά από εκτύπωση μηνύματος προειδοποίησης (warning). Ανάγνωση από μη υλοποιημένο τμήμα της μνήμης δίνει αποτέλεσμα 32'bx. Πόρτες: Dout: έξοδος δεδομένων (πλάτος 32 bits), Din: είσοδος δεδομένων (πλάτος 32 bits), Address: είσοδος διεύθυνσης λέξης (πλάτος 30 bits), RE: είσοδος επίτρεψης ανάγνωσης (1 bit), WE: είσοδος επίτρεψης εγραφής (1 bit), clk: ρολόϊ εγγραφής. Η ανάγνωση είναι ασύγχρονη, και γίνεται όποτε (RE & (~WE))==1. Η ανάγνωση αρχίζει με τη σταθεροποίηση των Address, RE, και WE, και δίνει αμέσως έγκυρα δεδομένα στην έξοδο Dout. Η εγγραφή είναι σύγχρονη, και γίνεται στη θετική ακμή του ρολογιού, όταν (WE & (~RE))==1.

Για απλότητα σχεδιάστε και περιγράψτε τη μονάδα ελέγχου (FSM, υπολογισμός σημάτων ελέγχου επομένου κύκλου, οδήγηση σημάτων ελέγχου παρόντος κύκλου) σε μορφή "συμπεριφοράς" (behavioral). Χρησιμοποιήστε συνδυαστική λογική (περιγραφόμενη με continuous assignments), if και case statements όπου χρειαστείτε, και κρατήστε την κατάσταση την FSM σε ακμοπυροδότητα flip-flops ή/και καταχωρητές (ετσι ώστε τα σήματα ελέγχου να αλλάζουν λίγο αργότερα από την ακμή του ρολογιού, διότι αλλιώς θα αντιμετωπίσετε πολλά προβλήματα χρονισμού!).

Προσέξτε ότι το module της μνήμης είναι οργανωμένο κατά λέξεις, και παίρνει διεύθυνση λέξης, ενώ ο επεξεργαστής, φυσικά, δουλεύει με διευθύνσεις bytes (byte-addressable). Η μετατροπή μεταξύ των δύο ειδών διευθύνσεων (και ο έλεγχος σφαλμάτων ευθυγράμμισης) είναι δική σας ευθύνη.

Η διεύθυνση του interrupt handler, στην οποία πηγαίνει αυτόματα το hardware σε περίπτωση εξαίρεσης, κανονικά είναι η 32'h40_00_00_00. Όμως, στον προσομοιωτή μας, όπου υλοποιούμε μόνο ένα πολύ μικρό κομμάτι μνήμης στις χαμηλές διευθύνσεις, δεν μπορούμε με αυτή τη διεύθυνση interrupt handler να δοκιμάσουμε τη λειτουργία των εξαιρέσεων (διακοπών). Γι' αυτό, ορίστε μιά δική σας παράμετρο (με 'define) που να είναι η διεύθυνση αυτή, και δώστε της όσο μικρή τιμή χρειάζεται γιά τους ελέγχους σας.