

Ερωτήσεις και απαντήσεις για το τμήμα του Dataflow

Ερώτηση 1. ALUControl των 3 bit γιατί με φτάνουν για να συμβολίσω τις 8 εντολές που ζητούνται. Υπάρχει σπάνιος τρόπος να συμβολίσω εγώ την κάθε εντολή ή κάνω αυθαίρετα την επιλογή μου π.χ. το ADD πρέπει οπωσδήποτε να είναι 000 ή το βάζω όπως θέλω;

Απάντηση: Μπορείτε να κάνετε όποια κωδικοποίηση θέλετε.

Ερώτηση 2. Σε κάθε αρχείο έχω τη δήλωση αρκετών generic. Η ανάθεση τιμής γίνεται στο generic map όταν δηλώνεται, σωστά; Άρα, αν κάποιος τρίτος θέλει να τα αλλάξει πρέπει να μπει σε κάθε αρχείο ξεχωριστά και να το αλλάξει, σωστά; Δεν υπάρχει τρόπος για global δήλωση;

Απάντηση: Όταν γίνεται κλήση του component μπορείτε να αλλάξετε τις τιμές generic με την εντολή generic map. Κοιτάξτε και στο pdf αρχείο **Κεφάλαιο 4: Γλώσσες περιγραφής υλικού (πλήρης έκδοση)**σελ. 288. Οπότε δεν χρειάζεται να το αλλάξετε μέσα στο αρχείο κάθε οντότητας.

Ερώτηση 3. Reset να βάλω σε όλες τις μονάδες για να πετάνε πάντα έξοδο 0 όταν αρχικοποιώ την εκτέλεση;

Απάντηση: Reset βάζετε μόνο σε καταχωρητές.

Ερώτηση 4. Για να δώσω στον πολυπλέκτη πριν το register file το σταθερό αριθμό 15, έκανα αυτό std_logic_vector(to_unsigned(15,RF_SIZE)) όπου RF_SIZE μια generic μεταβλητή που δηλώνει το μέγεθος του αρχείου καταχωρητών.

Απάντηση: Εξαρτάται από την υλοποίηση που έχετε αποφασίσει. Στην περίπτωση της άσκησης και της ARM αρχιτεκτονικής αυτό είναι κάτι σταθερό.

Ερώτηση 5. Στο INC4 αντιμετωπίζω την είσοδο ως unsigned ή signed; Εγώ το έχω βάλει ως unsigned επειδή ο PC είναι 0 ή θετικός.

Απάντηση: Εξαρτάται από την υλοποίηση που έχετε αποφασίσει. Σε κάθε περίπτωση θυμηθείτε ότι είναι καλό όλα τα in/out στα port να είναι είτε std_logic είτε std_logic_vector και να κάνετε εσείς τις όποιες μετατροπές χρειάζονται.

Ερώτηση 6. LDR,STR εντολές δε θα έχω να υποστηρίξω; Δεν αναφέρονται στις εντολές που πρέπει να υλοποιήσω;

Απάντηση: Στο αρχείο στο eclass υπάρχουν προς υλοποίηση ως εντολές μνήμης.

Ερώτηση 7. Ο assembler μου βγάζει το εξής στην εντολή LSL, ASR:

```
LSL R6,R5,#4;  
ASR R7, R6, #8;
```

αντί να μου βγάλει το #4 ως δυαδικό αριθμό 00100 μου βγάζει 01000(=8). Αντίστοιχα αν βάλω #5 μου βγάζει το 01010(=10). Γιατί συμβαίνει αυτό;

Απάντηση: Δεν τις έχω ελέγξει ακόμα τις συγκεκριμένες εντολές. Πιθανότατα το πιο δεξί Bit να ανήκει στο πεδίο **sh**.

Ερώτηση 8. Στην εκφώνηση αναφέρεται επίσης ότι «συμπεριλαμβάνει αθροιστή/αφαιρέτη των N bit, μονάδα λογικών πράξεων των N bit, μονάδα μεταφοράς δεδομένων

των N bit, μονάδα ολίσθησης των N bit». Αυτά πρέπει να υλοποιηθούν ως components, ή μας αρκεί απλώς να φτιάξουμε μια case όπου εξετάζουμε το ALUControl και πράττουμε ανάλογα, όπως παρακάτω;

Απάντηση: Εξαρτάται από την υλοποίηση που θα επιλέξετε.

Ερώτηση 9. Στην εκφώνηση λέει ότι για τις εντολές που θα υλοποιηθούν, χρειάζεται ALUControl των 3 bit. Οι εντολές όμως, είναι 13. Μήπως έχει γίνει λάθος και χρειάζονται 4 bit;

ή

Κάθε δυνατός συνδυασμός των 3bit του ALUControl, θα εκτελεί μια πράξη που αναγράφεται στο φυλλάδιο ή θα εκτελεί γενικότερες πράξεις και ένας συνδυασμός αυτών των γενικότερων πράξεων θα υπολογίζει μία από τις εντολές του φυλλαδίου (όπως π.χ. τα CMP,LDR,STR,B,BL);

Απάντηση: Οι εντολές ARM είναι 13 αλλά στην ALU υλοποιούνται συγκεκριμένες μόνο πράξεις. Οι εντολές με τις πράξεις δεν έχουν κάποια μοναδική αντιστοιχία. Το φυλλάδιο του project περιγράφει για κάθε εντολή ποια πράξη χρειάζεται και σε ποια bit της εντολής. Π.χ. για την BL χρειάζεται πρόσθεση. Οπότε η ALU εκτελεί ένα set πράξεων οι οποίες είναι αναγκαίες για την υλοποίηση των ARM εντολών. Η ALUControl χρειάζεται όντως 3 bit.

Ερώτηση 10. Οι πράξεις $MVN - S = 0$ και $ASR - S = 0$ σημαίνουν ότι έχουμε MVN και ASR χωρίς ενημέρωση των flags;

Απάντηση: Σωστά.

Ερώτηση 11. Ο mux πριν το Write Address του register file χρειάζεται για την εντολή BL;

Απάντηση: Φαίνεται ότι δεν χρειάζεται. Κοίτα και τη σελίδα 79 του Chapter 7.4. Γενικά η υλοποίηση του Datapath μπορεί να γίνει απλά από την απεικόνιση που προσφέρει το φυλλάδιο της άσκησης στη σελίδα 16.

Ερώτηση 12. Η εγγραφή ενός καταχωρητή στο αρχείο καταχωρητών γίνεται σύγχρονα ή ασύγχρονα;

Απάντηση: Φυλλάδιο της άσκησης, σελίδα 13, παράγραφος 2.2.1

Ερώτηση 13. Μπορεί να γίνει εγγραφή του R15 ή μόνο ανάγνωση του; Για παράδειγμα, όταν το Rd είναι ο R15, το αποτέλεσμα είτε από την ALU είτε από το Data Memory θα πάει ως καινούρια διεύθυνση στο PC reg. Ταυτόχρονα θα γίνει και κάποια εγγραφή στο R15 που υπάρχει στο Register file ή όχι;

Απάντηση: Φυλλάδιο της άσκησης, σελίδα 13, παράγραφος 2.2.1. Ο R15 **ΔΕΝ** υλοποιείται εντός του **Register File**, αν και δέχεται ως είσοδο την τιμή του R15. Θα πρέπει να δείτε τι τιμή έχει ο R15 στην αρχιτεκτονική ARM. Κοίτα και το σχήμα στη σελίδα 9 του Chapter 7.4.

Ερώτηση 14. Στο τέλος πως περνάμε το σήμα Result ως είσοδο στην αρχή? Γιατί όταν το κάνω το αρχικό μου σήμα μου βγάζει τιμές U.

Απάντηση: Αρχικά θα ήταν δυνατό να δώσουμε μια αρχική τιμή ώστε να αποφύγουμε το U. Το σήμα Result το οποίο είναι έξοδος από το component MemToReg θα είναι και είσοδος στα component PcSrc και στον mux πριν την είσοδο WD3 στο RegFile. Κοιτάμε λοιπόν αν οι είσοδοι στο MemToReg έχουν τιμές.

Ερώτηση 15. Αρχικά στις διαφάνειες μεθοδολογίας σχεδίασης ενός κύκλου αναφέρει ότι η μονάδα επέκτασης πρόσημου μπορεί να έχει και ως είσοδο "26 bit (άμεσος τελεστής των 24 bit πολλαπλασιασμένος επί 4)" , ποσά bit θα πρέπει να δεχτεί τελικά 26 ή 24, δηλαδή τον πολλαπλασιασμό επί 4 χρειάζεται να γίνει εκτός ή εντός της μονάδας επέκτασης πρόσημου;

Απάντηση: Στη σελίδα 73 του Chapter 7.4 αναφέρει ότι το imm24 έχει 24 bit. Η επέκταση στα 26 bit γίνεται λόγω της αρχιτεκτονικής ARM που έχει Words των 4 byte. Κοιτάξτε και του Chapter 6.2 σελίδες από 35-37. Όπως φαίνεται σε όλα τα διαγράμματα η είσοδος θα είναι των 24bit και η έξοδος στα 32bit.

Ερώτηση 16. Στις διαφάνειες μεθοδολογίας σχεδίασης ενός κύκλου στο κομμάτι περιγραφής των βασικών μονάδων της ALU , όσο αναφορά τη μονάδα μεταφοράς δεδομένων που είναι για τις εντολές MOV/MVN , υπάρχει κάπου κάποια αναφορά στις διαφάνειες ή κάποια πληροφορία γιατί με δυσκολεύει στο πως μπορούν να υλοποιηθούν οι εντολές αυτές ;

Απάντηση: Κοίτα την απάντηση στην ερώτηση 9.

Ερώτηση 17. θα χρειαστεί η ALU να έχει επιπλέον είσοδο shamt5 για εντολές ολίσθησης ή δεν χρειάζεται;

Απάντηση: Ναι, θα έχει επιπλέον είσοδο (πως αλλιώς θα γίνει η ολίσθηση;).

Ερώτηση 18. Στην αρχή του διαγράμματος, μετά τον καταχωρητή program counter ένα signal πηγαίνει στο instruction memory το οποίο είναι ένα ένας πίνακας τύπου ROM όπως έχει δοθεί από τις διαφάνειες. Σωστά;

Απάντηση: Σωστά. Και για την ακρίβεια είναι η διεύθυνση της μνήμης ROM που θα διαβαστεί και στην οποία υπάρχει η επόμενη προς εκτέλεση εντολή.

Ερώτηση 19. Από τον Program Counter ποιο μέρος του καταχωρητή θα πρέπει να δώσουμε σαν είσοδο στην ROM για να μας δώσει την αντίστοιχη έξοδο; Η να το πω διαφορετικά, ποιο byte του καταχωρητή PC θα πρέπει να είναι είσοδος στο Instruction memory;

Απάντηση: Το πλήθος φαίνεται φυλλάδιο της άσκησης, σελίδα 13, παράγραφος 2.1.2. Οπότε σίγουρα αρκούνε 6 bit, τα οποία περιγράφονται στη σελίδα 13.

Ερώτηση 20. Στο αρχείο pdf chapter 7.4. διαφάνεια 18 δίνει τον τύπο: $A[N-1:0] = PC[N+1:2]$. Μπορείτε να μας τον εξηγήσετε;

Απάντηση: Ως tip να πω ότι το Instruction Memory (IM) αποτελείται από λέξεις των 4 byte (32 bit κάθε γραμμή) ενώ ο PC δείχνει την τρέχουσα διεύθυνση σε byte. Άρα για να δείξει ο PC στην επόμενη γραμμή του IM πρέπει να αυξηθεί κατά 4 (να δείξει δηλαδή 4 byte μετά). Άρα πρακτικά ο PC δείχνει πάντα σε πολλαπλάσιο του 4 και άρα ο τύπος στην ερώτηση είναι πλέον προφανής.

Ερώτηση 21. Διαπίστωσα πως οι εντολές LSL-ASR και η MOV έχουν κοινό το πεδίο op(00) και το πεδίο cmd (1101). Ταυτόχρονα και το πεδίο S και το πεδίο I είναι κοινό όταν έχουμε πράξη μεταξύ registers. Αν πω να κοιτάξω το πεδίο sh τότε βλέπω πως στη MOV είναι πάντα 00, αλλά και στην LSL είναι 00. Να κοιτάξω το πεδίο shamt5 και να πω πως αν είναι διάφορο του μηδενός, τότε έχω shift? Αλλά δεν είναι ασφαλές αυτό, διότι μπορεί να γραφεί μια εντολή shift με τιμή shamt5=0. Πώς θα διαχωρίσω στο ποια εντολή λαμβάνω κάθε φορά;

Απάντηση: Η MOV μπορεί να αποτελεί μία υποπερίπτωση άλλης εντολής και να χειρίζεστε δύο εντολές ως μία.