

ΑΡΧΕΙΑ ΠΟΥ ΠΡΟΣΤΕΘΗΚΑΝ

1. **top.vhd** (Στον main φάκελο RISCV-32I-master) -> είναι το testbench για να τεστάρουμε τον επεξεργαστή, εκτός από το CLK(ανά 100ps ένας κύκλος ρολογιού) και το RESET έχω πρόσθεση και ένα σήμα cycle_counter που μας λέει σε ποιον κύκλο ρολογιού βρισκόμαστε.
2. **testB.vhd** (Στον main φάκελο RISCV-32I-master) -> είναι το testbench για να ελέγξουμε αν λειτουργεί σωστά ο πολλαπλασιαστής(MUL)
3. **MUL.vhd** (Στον φάκελο TOOLBOX Components) ->είναι ο πολλαπλασιαστής.

ΑΡΧΕΙΑ ΠΟΥ ΑΛΛΑΧΘΗΚΑΝ

Για να προσθέσω την εντολή του πολλαπλασιασμού αύξησα το control word από 20 σε 21 bits, ώστε το ALU OP να χωράει και την εντολή του πολλαπλασιασμού. Έτσι το ALU OP έγινε 3 bits (000: ADD, 001: SUB , 010 : LOGIC OPERATION , 011 : SHIFT OPERATION , 100 : MUL), για να γίνει αυτό έκανα τις εξής αλλαγές:

1. **TOOLBOX.vhd** (Στον main φάκελο RISCV-32I-master) ->
 - a. πρόσθεσα το component του πολλαπλασιαστή(MUL.vhd) {σειρά 427..436}
 - b. άλλαξα το input MUX_2X1_SEL σε MUX_4X1_SEL με μέγεθος (1DOWNT0 0) στο component ID_DECODER {σειρά 173}
 - c. άλλαξα το μέγεθος των CTRL_WORD απο (17 DOWNT0 0) σε (18 DOWNT0 0) {σειρά 292} και Το_EXE_ALU απο (8 DOWNT0 0) σε (9 DOWNT0 0) {σειρά 295} στο component CONTROL_WORD_REGROUP
2. **INSTRUCTION_DECODE.vhd** (Στον φάκελο PIPELINE Components) ->
 - a. άλλαξα το μέγεθος του FUNCT7 απο 1 bit σε 2 bits {σειρά 63}
 - b. άλλαξα τα bits που πέρνει το FUNCT7 απο το IF_WORD απο IF_WORD(30) σε IF_WORD(31 DOWNT0 30) {σειρα 83}
 - c. άλλαξα το input MUX_2X1_SEL,που πέρνει το component ID_DECODER, σε MUX_4X1_SEL {σειρα 88}
3. **ID_DECODER.vhd** (Στον φάκελο TOOLBOX Components) ->
 - a. άλλαξα το input MUX_2X1_SEL σε MUX_4X1_SEL με μέγεθος (1 DOWNT0 0) {σειρα 61}
 - b. πρόσθεσα το SIGNAL R_MUL με τιμή ("111"&"1"&"0"&"111"&"00"&"100"&"0"&"0"&"0"&"0"&"0"&"0"&"0"&"0"&"0")
 - c. πρόσθεσα 1 επιπλέον bit σε όλα τα control words στην θέση 10 αφού πλέον το ALU OP του CTR_WD είναι 3 bits και όχι 2
 - d. άλλαξα όλα τα components MUX2X1 σε MUX4X1, ώστε να μπορέσω να προσθέσω και το control word R_MUL {σειρές 129..159}
 - e. άλλαξα το input sel που πέρνουν όλα τα MUX4X1 components απο MUX_2X1_SEL σε MUX_4X1_SEL {σειρές 137,147,157}
 - f. πρόσθεσα το R_MUL στο MUX4X1 {σειρα 145}

4. **CONTROL_WORD_REGROUP.vhd** (Στον φάκελο TOOLBOX Components) ->
 - a. άλλαξα το μέγεθος των CTRL_WORD{σειρα 7} απο (17 DOWNT0 0) σε (18 DOWNT0 0) TO_EXE_ALU{σειρα 10} απο (8 DOWNT0 0) σε (9 DOWNT0 0)
 - b. άλλαξα τα bits που πέρνει το TO_EXE_ALU απο [[CTRL_WORD(12 DOWNT0 11) & CTRL_WORD(10 DOWNT0 9) & CTRL_WORD(8) & CTRL_WORD(7) & CTRL_WORD(6) & CTRL_WORD(5) & CTRL_WORD(1)]] σε [[CTRL_WORD(13 DOWNT0 12) & CTRL_WORD(11 DOWNT0 9) & CTRL_WORD(8) & CTRL_WORD(7) & CTRL_WORD(6) & CTRL_WORD(5) & CTRL_WORD(1)]] και TO_OTHERS απο [[CTRL_WORD(16) & CTRL_WORD(15 DOWNT0 13) & CTRL_WORD(17)]] σε CTRL_WORD(17) & CTRL_WORD(16 DOWNT0 14) & CTRL_WORD(18)
5. **EXE.vhd** (Στον φάκελο PIPELINE Components) ->
 - a. άλλαξα το μέγεθος του OP απο (8 DOWNT0 0) σε (9 DOWNT0 0) {σειρα 31}
 - b. άλλαξα το bit που πέρνει το input SEL στο component MUX2X1_BIT απο OP(8) σε OP(9) στις {σειρες 68,75}
 - c. άλλαξα το bit που πέρνει το input SEL στο component EXE_LOGIC_MODULE απο OP(8 DOWNT0 7) σε OP(9 DOWNT0 8) {σειρα 117}
 - d. άλλαξα το bit που πέρνει το input SEL στο component BARREL_SHIFTER απο OP(8 DOWNT0 7) σε OP(9 DOWNT0 8) {σειρα 126}
 - e. πρόσθεσα ένα Signal MUL_RES που αποθηκεύει το αποτέλεσμα του πολλαπλασιαστή(MUL) {σειρα 54}
 - f. πρόσθεσα ένα component MUL που πέρνει ως inputs το A και B κάνει την πράξη του πολλαπλασιασμού και αποθηκεύει το αποτέλεσμα στο signal MUL_RES {σειρα 89..95}
 - g. άλλαξα το ALU_MUX component απο MUX4X1 σε MUX8X1 και τα bits που πέρνει το SEL απο OP(6 DOWNT0 5) σε OP(7 DOWNT0 5) και πρόσθεσα σαν D4 το MUL_RES {σειρα 147..160}
 - h. πρόσθεσα ένα CONSTANT GND που περιέχει 32 bits με 0 {σειρά 59}
6. **PIPE_ID_TO_EXE_REGISTER.vhd** (Στον φάκελο PIPELINE Components) ->
 - a. άλλαξα το μέγεθος των I_CTRL_WORD{σειρα 34} O_CTRL_WORD{σειρα 47} BUF_CTRL_WRD{σειρα 66} απο (17 DOWNT0 0) σε (18 DOWNT0 0)
 - b. πρόσθεσα 1 επιπλέον bit σε όλα τα BUF_CTRL_WRD στην θέση 10 αφού πλέον το ALU OP είναι 3 bits και όχι 2
7. **PIPELINE.vhd** (Στον main φάκελο RISCV-32I-master) ->
 - a. άλλαξα το μέγεθος των I_CTRL_WORD{σειρα 61} O_CTRL_WORD{σειρα 75}
 - b. άλλαξα το μέγεθος του OP απο (8 DOWNT0 0) σε (9 DOWNT0 0) στο component EXE {σειρα 177}

8. **RV32I.vhd** (Στον main φάκελο RISCv-32I-master) ->
- a. άλλαξα το μέγεθος των GENERIC MAP που δίνουμε στο component INSTRUCTION_DECODE, το CTRL_WORD_TOTAL απο 20 σε 21 και το CTRL_WORD_OUT απο 18 σε 19 {σειρά 216}
 - b. άλλαξα το μέγεθος των I_D{σειρα 31},PIPE_B_OUT_CTRL_WORD{σειρα 85},ID_OUT_OPCODES{σειρα 124} απο (17 DOWNT0 0) σε (18 DOWNT0 0)
 - c. άλλαξα το μέγεθος του ALU_OPCODE απο (8 DOWNT0 0) σε (9 DOWNT0 0) {σειρα 142}

Μετά από όλες αυτές τις αλλαγές λειτουργεί κανονικά η εντολή MUL:

Π.Χ. Για την εντολή MUL \$t3,\$t4,\$t5 ($t3 = t4 * t5$) :

Funct7	rs2	rs1	funct3	rd	opcode
1000000	11110	11101	000	11100	0110011 ή 81df0e33