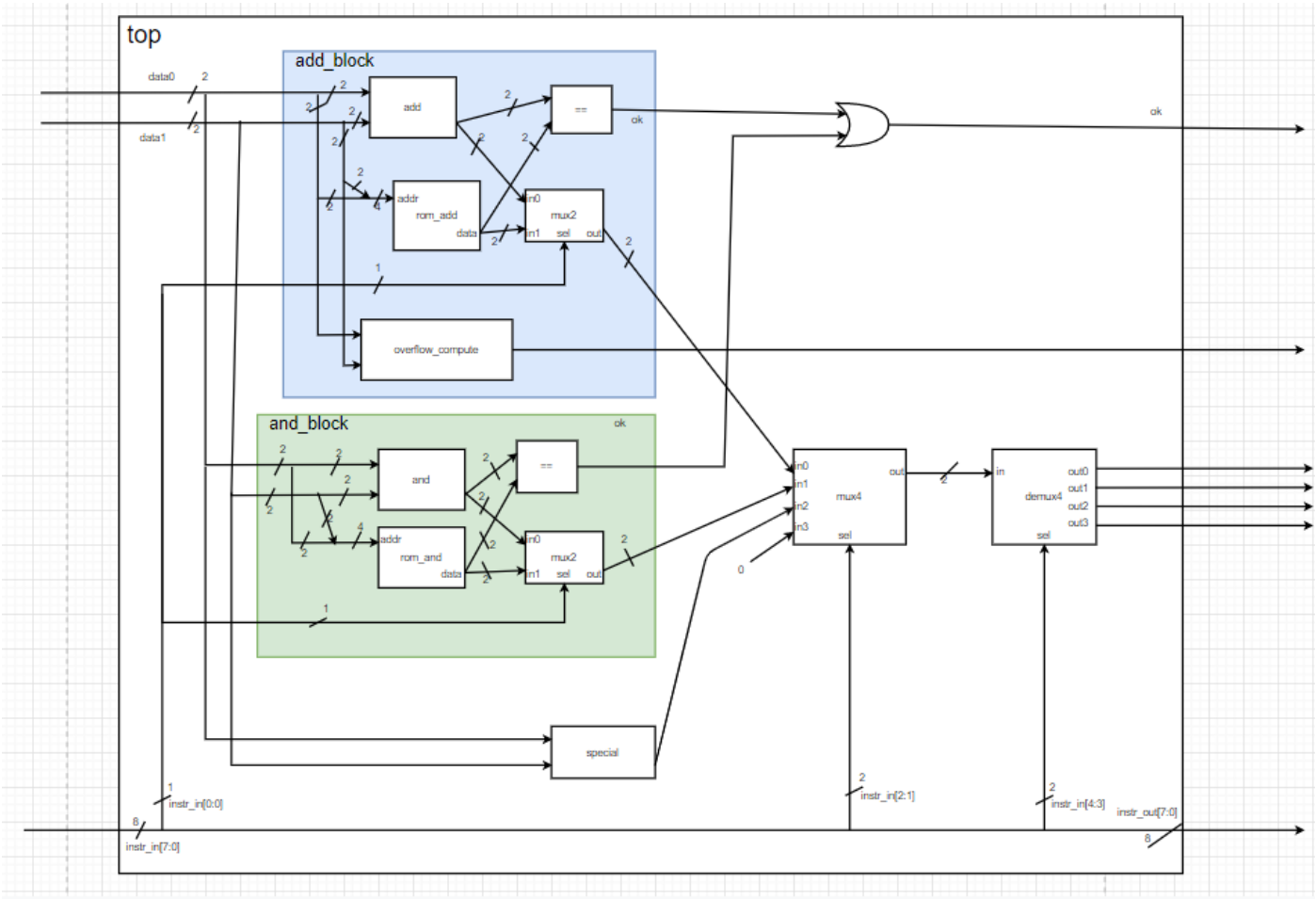
Subiect:

Timp de lucru: 1.20h

Cerinta:

Implementati schema de mai jos in verilog si efectuati simulari pentru verificarea functionalitati.

Schema bloc:



Descrierea schemei + cerinte speciale:

Circuitul de mai sus este folosit pentru a testa cele 2 memorii rom: rom_add si rom_and. Aceste 2 memorii sunt folosite pe post de look-up tables pentru cele 2 operatii, adunare si "si" pentru intrari pe 2 biti. Acesta este alcatuit din 2 blocuri de calcul (+ un bloc de calcul special) impreuna cu un multiplexor ce selecteaza care din calcule vor ajunge la iesire si un demultiplexor ce selecteaza iesirea.

Firul de ok verifica functionarea corecta a modulelor de tip rom comparat cu cele de calcul standard, prin compararea rezultatelor acestora. Acesta ar trebuii sa fie mereu "1".

Flag-ul de overflow ar trebuii sa fie 1 atunci cand suma nu incape pe 2 biti.

Ambele memorii rom sunt folosite ca look-up table pentru operatiile respective, astfel pentru fiecare adresa, iesirea ar trebuii sa fie rezultatul operatiei corespunzatoare, avand datele de la intrare cei 2 + 2 biti care alcatuiesc adresa.

Modulul "==" este un comparator de egalitate.

Cele 2 blocuri: add_block si and_block sunt module separate ce se instantiaza apoi in top.

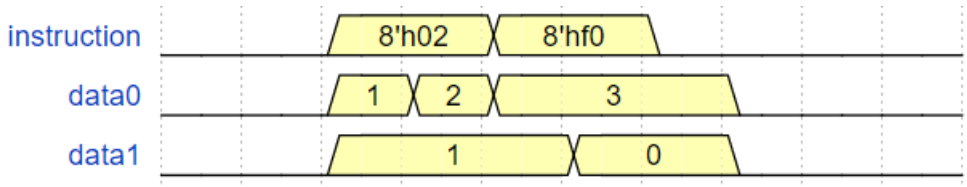
Modulul "special" : intoarce 1 daca cele 2 numere de la intrare au numar egal de biti de 1 sau 0 otherwise.

Dati denumiri logice si sugestive oriunde acestea nu sunt specificate in design.

Adaugati dimensiunile corespunzatoare firelor, acolo unde considerati ca ar trebuii.

Pentru simulare:

part 1) Formele de unda pentru intrare trebuie corespunda cu figura de mai jos:



Spatiul dintre liniile punctate (verticale) reprezinta 5 nano secunde.

part 2) testati functionarea flag-ului de overflow.

Simularea se inchide dupa 200 de unitati de timp.

Barem:

Total - 20p

design - 14p

- top - 2
- add_block - 1
- and_block - 1
- demux4 - 1.5
- mux4 - 1
- special - 2
- mux2 - 0.5
- rom_add - 1.5
- rom_and - 1.5
- "==" - 0.5
- overflow_compute - 0.5
- and + add = 1

simulare - 6p

- testbench 4p - 1p instantiere + 2p generarea corecta a semnalelor (1.5 fiecare parte).
- demonstratie simulare: 2p - poza forme de unda cu toate modulele puse in simulare ca grupuri si vizibile semnalele din tb, intrari si iesiri. Salvati setarile ca sa pot si eu sa le vad usor dupa.