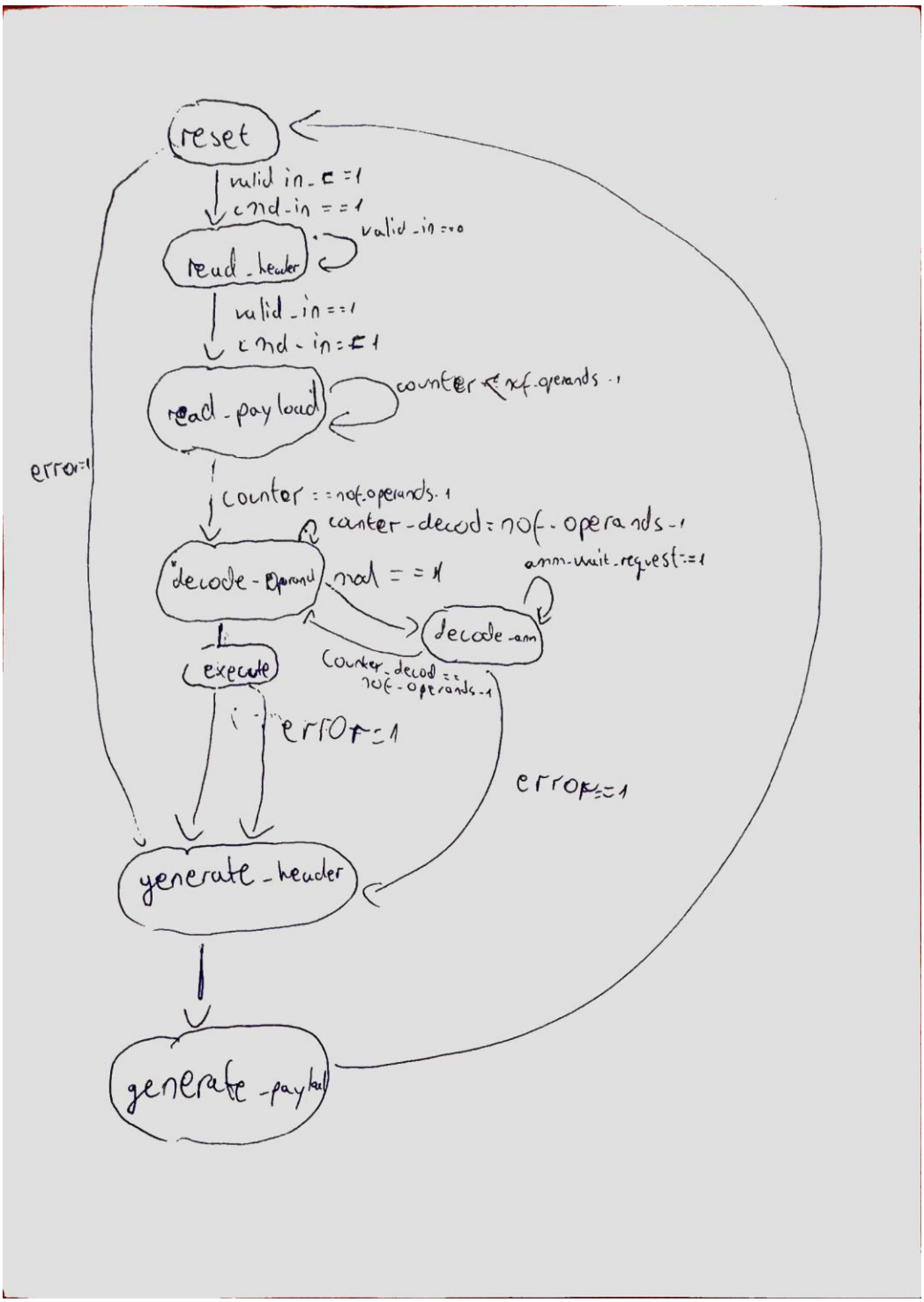
Ghideu Andrei
333 AB

Am implement in modulul alu un automat care primește date, le decodifică, efectuează operații și întoarce rezultatele respectând același protocol.



PS: am făcut o greseală in desen, daca am error ==1 plec in generate_header din read_header

Ghideu Andrei
333 AB

Ideea automatului meu este următoarea: In starea mea initiala ***`reset*** atribui valoarea 0 pentru toate reg-urile pe care urmeaza sa le utilizez in starile urmatore si celalalt scop este de a ma intoarce in reset pentru urmatorul pachet de date.

In `read_header citesc informatii legate de header urmarind protocolul pentru a determina codul operatiei si numarul de operanzi. Daca intalnesc numar nul de operanzi, sar direct la starile de output pentru a transmite datele de iesire deoarece nu are rost sa intru in `read_payload si execute fara operanzi. Daca nu primesc semnal pozitiv de la valid_in ma intorc in aceeasi stare pentru a citi date.

In `read_payload introduc toate payload-urile intr-un buffer pentru a le prelucra in starea urmatoare.

In starile de decode, ma concentrez asupra modului de adresare, separand adresarea imediata prin `decode_operand si adresarea indirecta prin `decode_amm. In `decode_operand introduc date direct intr-un vector de valori, operands_addr, si verific daca mai am operanzi de decodificat. In aceasta verificare si in cea urmatoare ei, `decode_amm , nu tin cont de cont de modul de adresare in cautarea mea.

In `decode_amm lucrez cu memoria. Acest task este mai dificil deoarece trebuie sa astept minim 3 ciclii de ceas. Incep operatia de citire prin amm_read = 1, pentru a schimba amm_waitrequest in 0 la un moment dat, el fiind mereu 1 fara a schimba amm_read. Daca waitrequest ramane activ ma intorc in aceeasi stare. Aceasta bucla de intoarcere in aceeasi stare tine locul unei instructiuni repetitive. Cand waitrequest este 0, verific amm_response. In situatia in care acest semnal este 0, poti introduce date in vectorul meu din adresa de memorie, altfel am cazuri de eroare. Pasul final este sa pun amm_read pe 0 pentru a schimba waitrequest pe 1, pregatindu-ma pentru o noua citire din memorie daca exista.

In starea de `execute calculez operatiile si actualizez rezultatul in situatia in care am mai multi operanzi.

In `generate_output_header si `generate_output_payload generez pachetul de iesire. In situatia unei erori, rezultatul meu in payload_out va intoarce 0xbad.