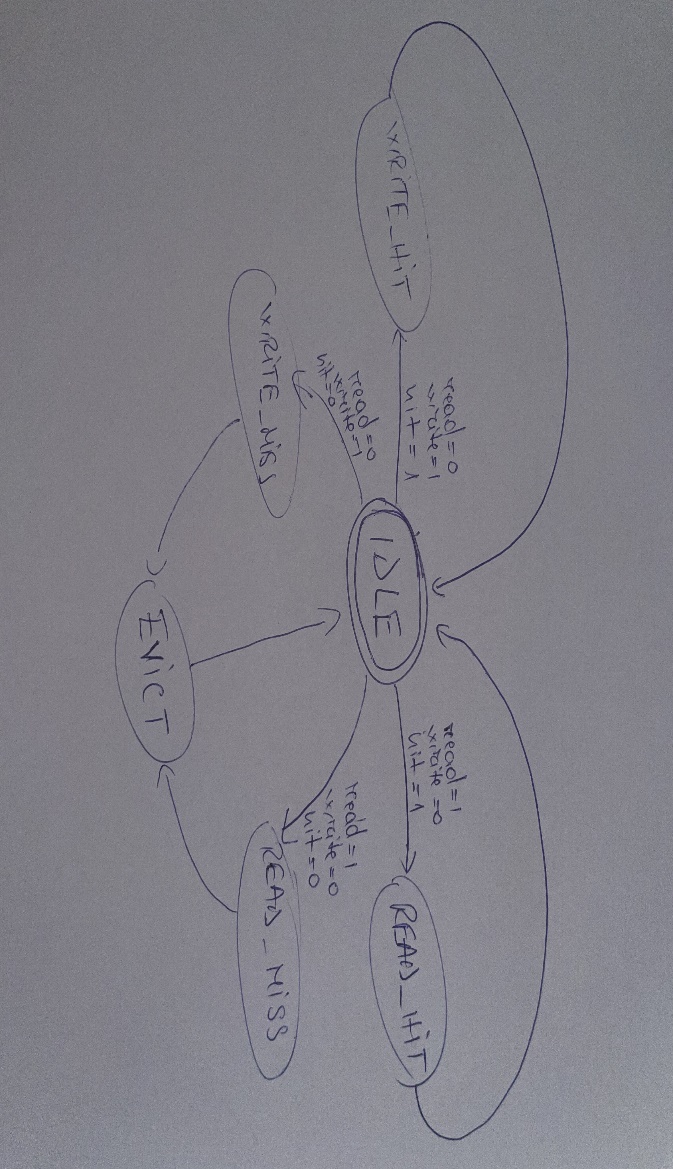
Cache controller

Echipa: David Darius, David Raul-Bogdan, Dobridoreanu Daniel

Specificatii de design

Controllerul de cache implementat este un 4 way set-associative cache, cu o dimensiune totală de 32 KB și blocuri de 64 bytes fiecare. Controllerul gestionează operațiile de citire și scriere utilizând un automat cu stari finite(FSM) și implementează un replacement policy Least Recently Used (LRU). Cache-ul este organizat în 128 de seturi, fiecare set având 4 căi (blocuri).



*Reprezentare grafica FSM*

După cum putem vedea, automatul incepe in starea IDLE. În funcție de input-urile read și write, dar și de variabila hit care rezultă din compararea tag-urilor și a bitului de valid în modulul tag\_array, automatul trece în stările de hit (WRITE\_HIT sau READ\_HIT) când hit este 1 sau în stările de miss (WRITE\_MISS sau READ\_MISS) când hit este 0. În urma fiecărui miss, trecem în starea de EVICT și facem schimbările necesare la nivelul memoriei. Din starea EVICT ne vom întoarce în IDLE.

A diagram of a computer

Description automatically generated

*Diagrama functionala a cache controllerului*

În Verilog, am implementat urmatoarele module: cache\_controller, tag-array, data\_array, lru\_module și fsm\_module.

**Modulul tag\_array** gestionează stocarea și verificarea tag-urilor pentru blocurile de date din cache. Fiecare set din cache conține un număr de tag-uri corespunzător nivelului de asociativitate al cache-ului.

În fiecare ciclu de clock, modulul verifică dacă tag-ul curent se potrivește cu oricare dintre tag-urile stocate în setul specificat (set\_index) și dacă acel tag este valid. Dacă există o potrivire, semnalul hit este setat la 1 și way indică calea corespunzătoare.

A computer screen with white text

Description automatically generated

Dacă write\_enable este activat, tag-ul și bitul de valid sunt actualizate pentru calea specificată de update\_way.

**Modulul data\_array** gestionează stocarea și accesarea efectivă a datelor în cache. Fiecare bloc de date este accesat printr-un set de indecsi care specifică setul și calea.

În funcție de semnalul read sau write, modulul citește sau scrie datele din/în cache. În base\_index se calculează indexul de bază pentru linia curentă din cache. În word\_offset determinăm offset-ul word-ului în cadrul unui bloc.

Dacă read este activ, datele sunt citite din data\_array și plasate pe read\_data.

Dacă write este activ, datele din write\_data sunt scrise în data\_array.

A screen shot of a computer program

Description automatically generated

**Modulul lru\_module** gestionează replacement policy-ul Least Recently Used (LRU), urmărind care blocuri din cache au fost utilizate recent.

Modulul ține evidența utilizării fiecărui bloc din cache printr-un contor LRU. La fiecare acces, contorul LRU este actualizat pentru a reflecta utilizarea curentă. update\_lru este folosit pentru a decide care bloc să fie înlocuit în cazul unui miss.

A computer code with white text

Description automatically generated

**Modulul fsm\_module** este un automat cu stări finite care gestionează tranzițiile între diferitele stări ale controllerului de cache.

Modulul gestionează tranzițiile de stare pe baza semnalelor de intrare și a stării curente. Stările FSM includ: IDLE, READ\_HIT, READ\_MISS, WRITE\_HIT, WRITE\_MISS, EVICT. În funcție de starea curentă și de semnalele de intrare, FSM decide următoarea stare și semnalele de control corespunzătoare.

**Modulul cache\_controller** integrează toate celelalte module și gestionează operațiile de citire și scriere în cache, asigurându-se că datele sunt accesate corect și că stările sunt actualizate corespunzător.

Modulul descompune adresa în componentele sale (set index, tag, block offset).

A computer screen shot of white text

Description automatically generated

În funcție de semnalele de control (read, write), inițiază operațiile corespunzătoare în modulele tag\_array și data\_array. fsm\_module gestionează tranzițiile de stare și semnalele de control pentru operațiile de citire și scriere. lru\_module actualizează politica LRU.