**Experiment 9**

**Direct mapped cache design**

'Direct Mapped Cache' component in the 'other components' drawer in the simulator supports both writing in the cache and the cache mapping. No replacement policy has been implemented. Initially the cache is empty, user has to give inputs. the component contains **4 sets, each set has 5 bits, the left most bit is the valid bit, next 2 bits are tags, next bits are data bits**, also it contains a **one dimensional array of memory with 4 bit to store the memory address**, user has to give this address input also

|  |  |  |
| --- | --- | --- |
| Tag | Block no. | word |

**Loading data in the cache**

* global initialisation: **(S=1, R/W'A=0, R/W'D=0, Den=1)**
* in cache line 0, load as follows:
  + data= "11" (D1=1, D0=1)
  + tag= "10" (T0=0, T1=1)
  + valid bit= "1" (valid=1)

**Examining hit behaviour**

* load data in address latch as: **( Set S=1, R/W A=1,R/W D=0, Den=1)**
  + set: "00" (A0=0, A1=0),
  + tag= "10" (A3=1, A2=0)

A0 A1 A2 A3

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 0 | 0 | 1 |

SET TAG

* **initiate cache mapping:**
  + **S=0, R/W'A=1, R/W'D=1, Den= 0**
* check output:
  + F0=1, F1=1, hit/miss=1

**Examining miss behaviour due to mismatch of tag:**

* load data in address latch as:
  + set: "00" (A0=0, A1=0),
  + tag= "11" (A3=1, A2=1)

A0 A1 A2 A3

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 0 | 1 | 1 |

* initiate cache mapping:
  + S=0, R/W'A=1, R/W'D=1, Den= 0
* check output:
  + F0=0, F1=0, hit/miss=0

1. For a 'Direct Mapped Cache' component pin configuration is:
   * pin-32= S (selects whether user wants to perform cache write or cache mapping) for **Cache write S=1, Cache mapping S=0**
   * pin-31= **R/W'A (selects whether user wants to input the address or cache mapping)**
   * pin-30=A3, pin-29=A2, pin-28=A1, pin-27=A0 (these 4 pins are used to give address input). A3 is the most significant bit and A0 is the least significant bit. **A3 and A2 will be compared with the tag. A1 and A0 will select the corresponding set**.
   * pin-26**= R/W'D(selects whether user wants to input in the set of cache or cache mapping**)
   * **pin-25= M1, pin-24=M0** (M1 is the most significant bit and M0 is the least significant bit). these **two bits are used for cache write purpose, it selects the particular set of which user wants to give inputs** to the valid bit, tag bits and data bits.
   * pin-23= Den (this is an enable input which has to set for any write purpose in the cache).
   * pin-21= valid bit
   * pin-20= T1, pin-19=T0 (T1 is the most significant bit and T0 is the least significant bit). These are tag bits.
   * pin-18= D1, pin-17=D0 (D1 is the most significant bit and D0 is the least significant bit). These are data bits.
   * pin-14= Hit/Miss bit (if it gives 1 then hit otherwise miss)
   * pin-15= F1, pin-16=F0 (F1 is the most significant bit and F0 is the least significant bit). These are output data bits and will be given only when there is a hit.
2. **Essential pin configurations for writing in the cache: S=1, R/W'A=0, R/W'D=0, Den= 1**
3. **Essential pin configurations for cache mapping: S=0, R/W'A=1, R/W'D=1, Den= 0**

