به نام خدا

گزارش پروژه معماری کامپیوتر

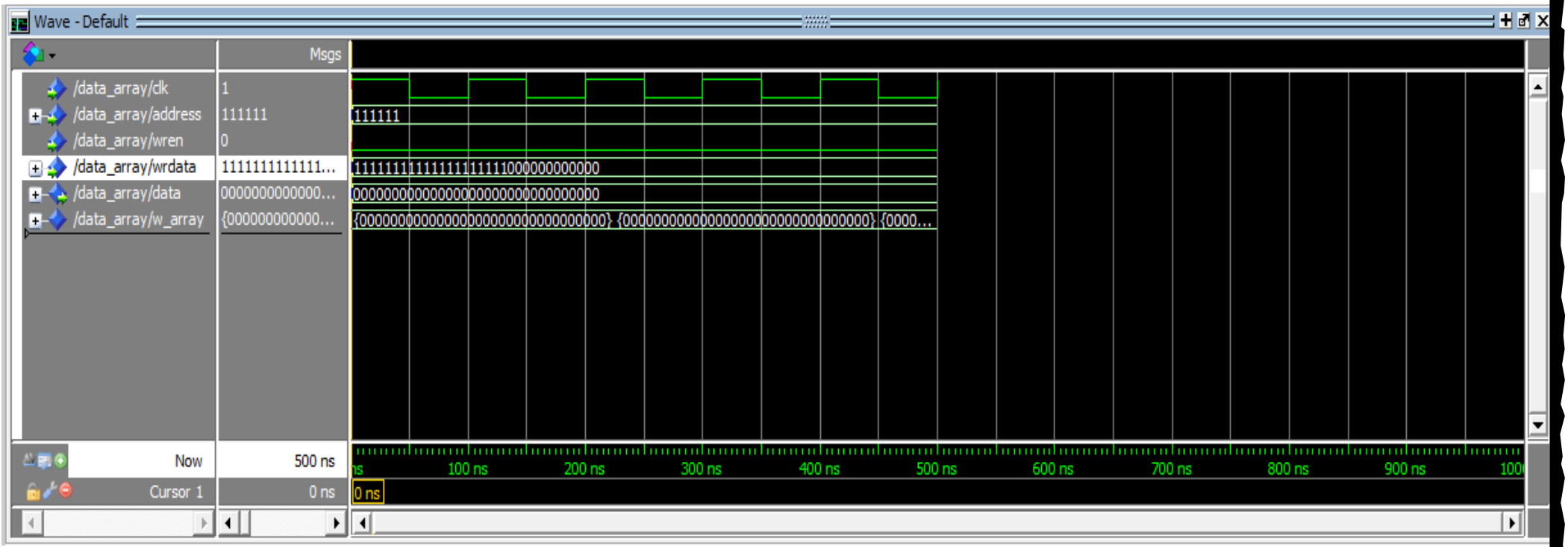
محمدمهدی آقاجانی

9331056

آرایه داده ای (data array)

آرایه ای از std logic vector ها ست که در واقع 64 عنصر دارد. این ماژول حساس به لبه بالارونده کلاک است و اگر سیگنال wren فعال باشد اقدام به نوشتن می کند در غیر این صورت از همین آرایه ، داده مربوطه را می خواند و در خروجی قرار میدهد.

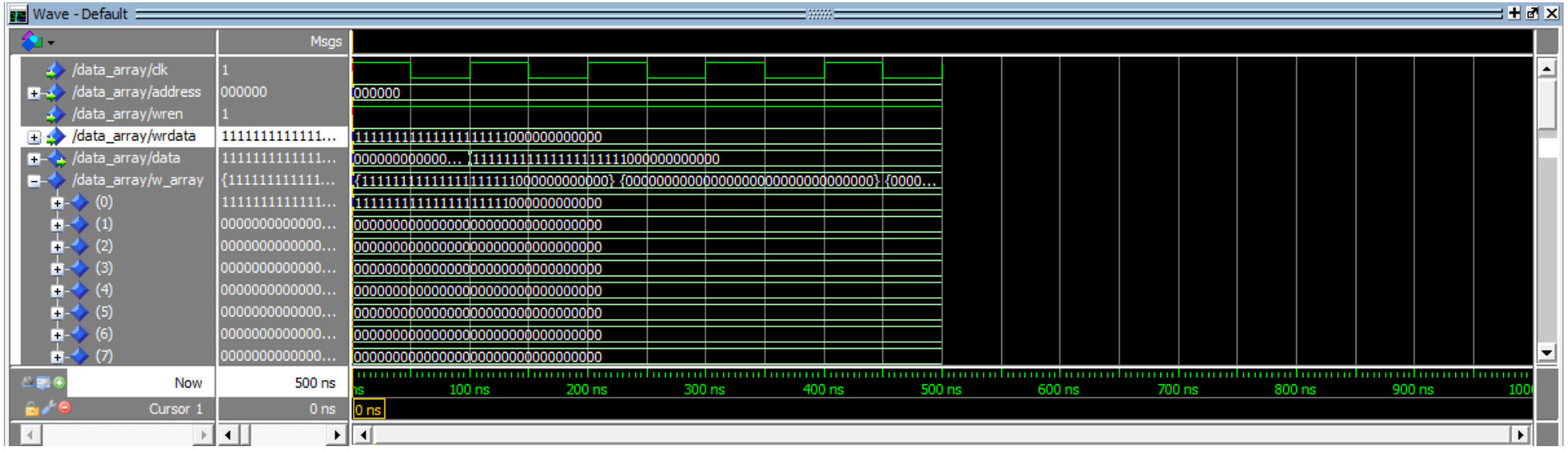
شکل موج خروجی وقتی که wren = 0 :



در این حالت داده در آدرس 111111 که در واقع 00000000000000000000000000000000 است در خروجی قرار گرفته است :

شکل موج خروجی وقتی wren = 1 است :

در این حالت داده wrdata در خانه آدرس که 000000 است قرار گرفته همچنین به خاطر این جایگزینی موفق داده در خروجی نیز قرار گرفته است :

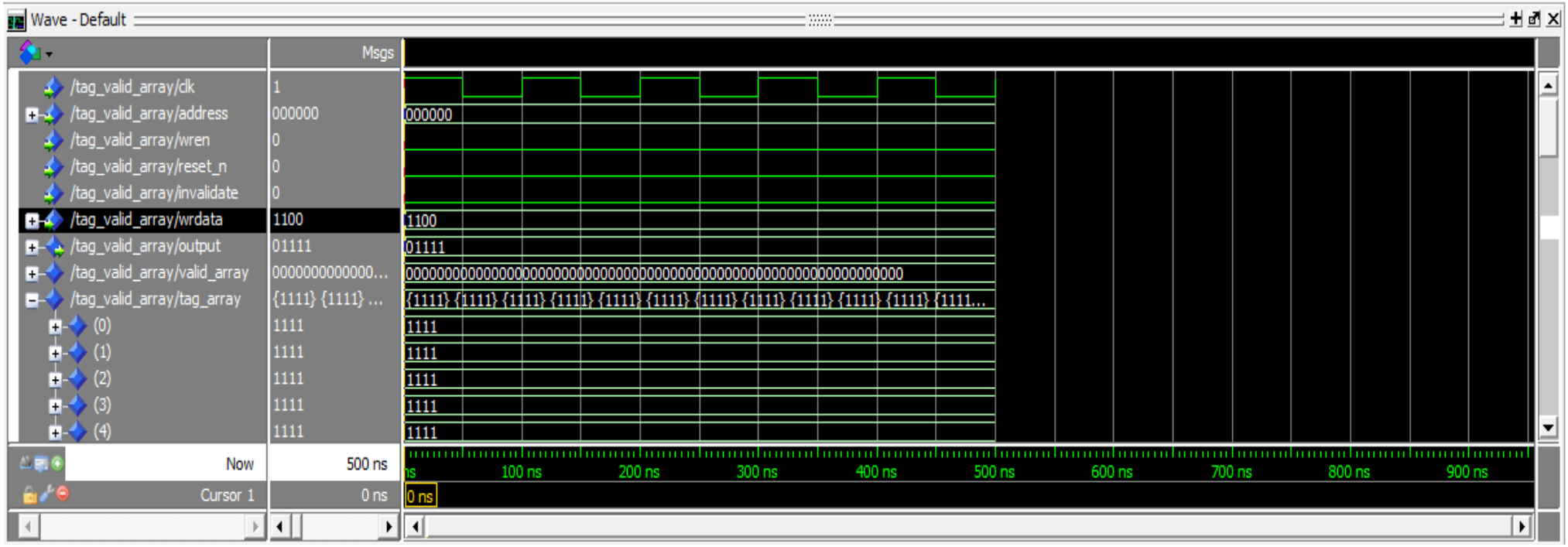


Tag valid array :

نحوه ساخت این ماژول مانند آرایه داده ای می باشد و حساس به لبه بالارونده کلاک است توجه کنید که در ابتدا به صورت پیش فرض تمامی داده های tag ها برابر 1111 تمامی داده های valid برابر صفر قرار داده شده است

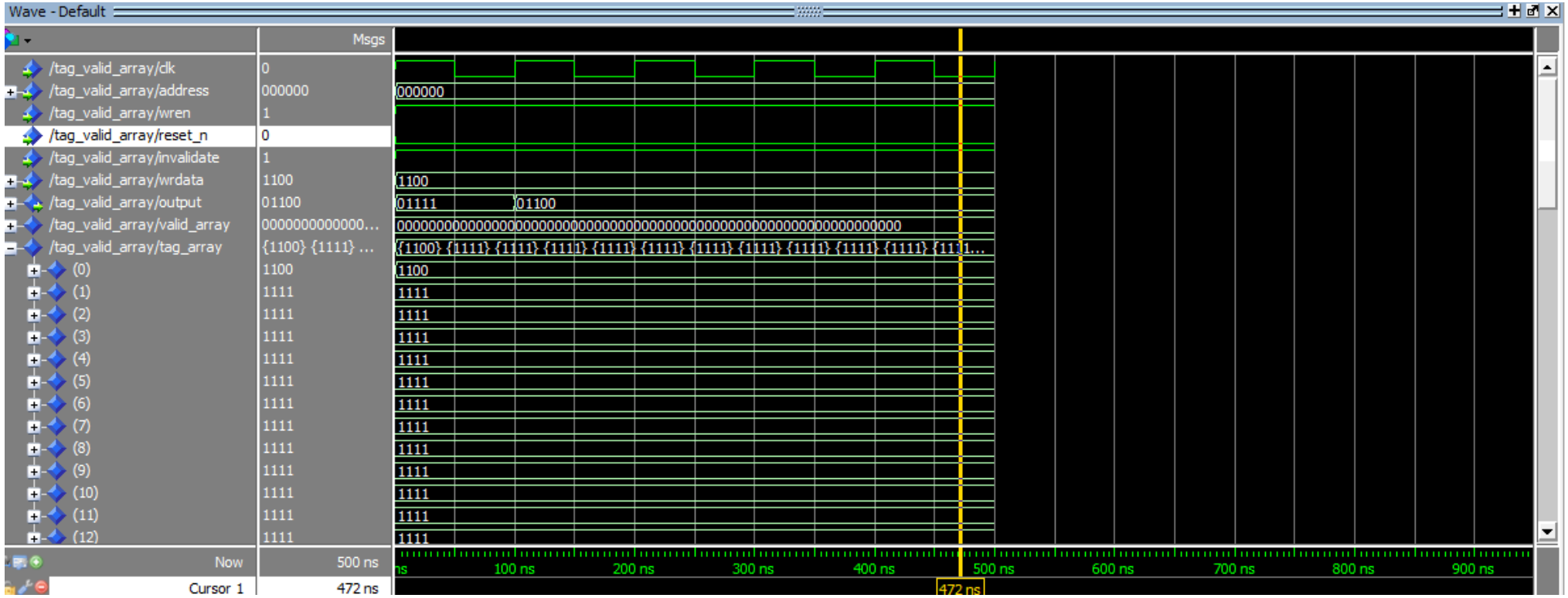
در زیر شکل موج به ازای reset = 0 , wren = 0 را مشاهده می کنید :

در این حالت بدلیل اینکه سیگنال invalidate برابر صفر است valid خانه مورد نظر تغییری نکرده و برابر همان مقدار قبلی ست به همین دلیل در خروجی سیگنال 01111 را مشاهده میکنیم



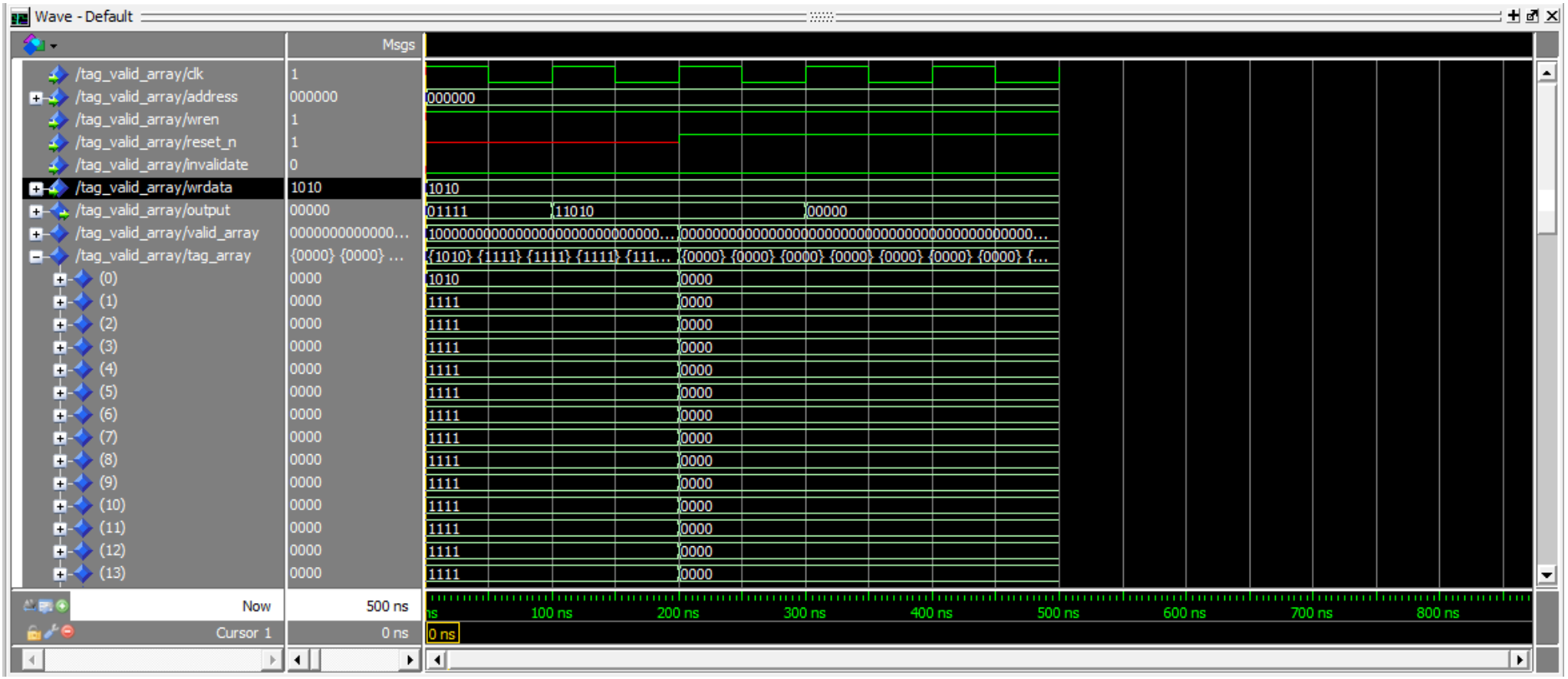
در این حالت شکل موج را به ازای wren = 1 , reset = 0 مشاهده می کنید

دقت کنید که به این علت valid مربوط به آدرس داده شده صفر شده است که invalidate برابر 1 در نظر گرفته شده و UI باید valid مربوط به آدرس داده شده برابر یک قرار میگرفت



در این حالت reset برابر یک قرار داده شده :

دقت کنید که wren برابر یک است و invalidate برابر صفر است به همین خاطر در ابتدا خروجی برابر داده نوشته شده در آرایه قرار میگیرد و سپس بعد از اینکه ریست برابر یک قرار داده می شود تمامی آرایه برابر صفر قرار میگیرد



منطق مطابقت (miss-hit logic) :

این مدار چون حساس به کلاک نیست باید در سطح گیت پیاده شود که از گیت xnor برای مقایسه استفاده شده است و در نهایت با valid مقایسه می شود تا نتیجه را مشخص کند.

در زیر جدول درستی و جدول کارنو را می کشیم . توجه کنید که برای هر بیت ورودی یک جدول می کشیم و بعد آن ها را ادغام میکنیم. این کار بدین دلیل است که تعداد ورودی های بیتی خیلی زیاد می باشد و جدول کارنو دچار پیچیدگی می شود.

Tag(i)	W(i)	valid	Equal_w(i)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

10	11	01	00	valid
0	0	0	0	**0**
0	1	0	1	**1**

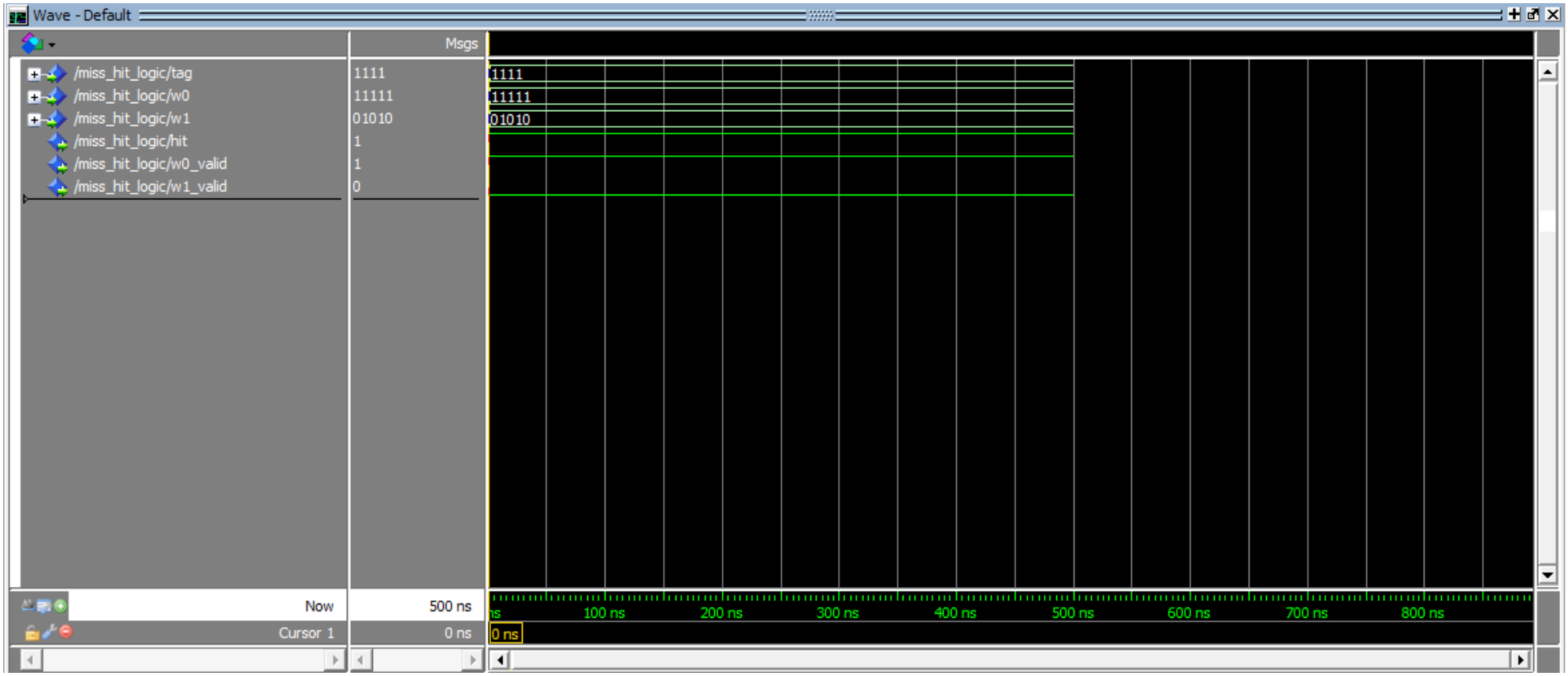
با توجه به جدول بالا خواهیم داشت :

$$equal_{w(i)} = valid.\sim w(i).\sim tag(i) + tag(i).w(i).valid = valid.(tag(i) \odot w(i))$$

با توجه به بالا و با توجه به این نکته که باید برای اینکه یک مطابقت رخ دهد تمامی equal_w(i) ها برابر یک باشند پس خواهیم داشت :

$$w_{valid} = valid.\big(tag(0) \odot w(0)\big).\big(tag(1) \odot w(1)\big).\big(tag(2) \odot w(2)\big).\big(tag(3) \odot w(3)\big)$$

در این حالت شکل موج را در حالتی که hit رخ داده است مشاهده می کنید:



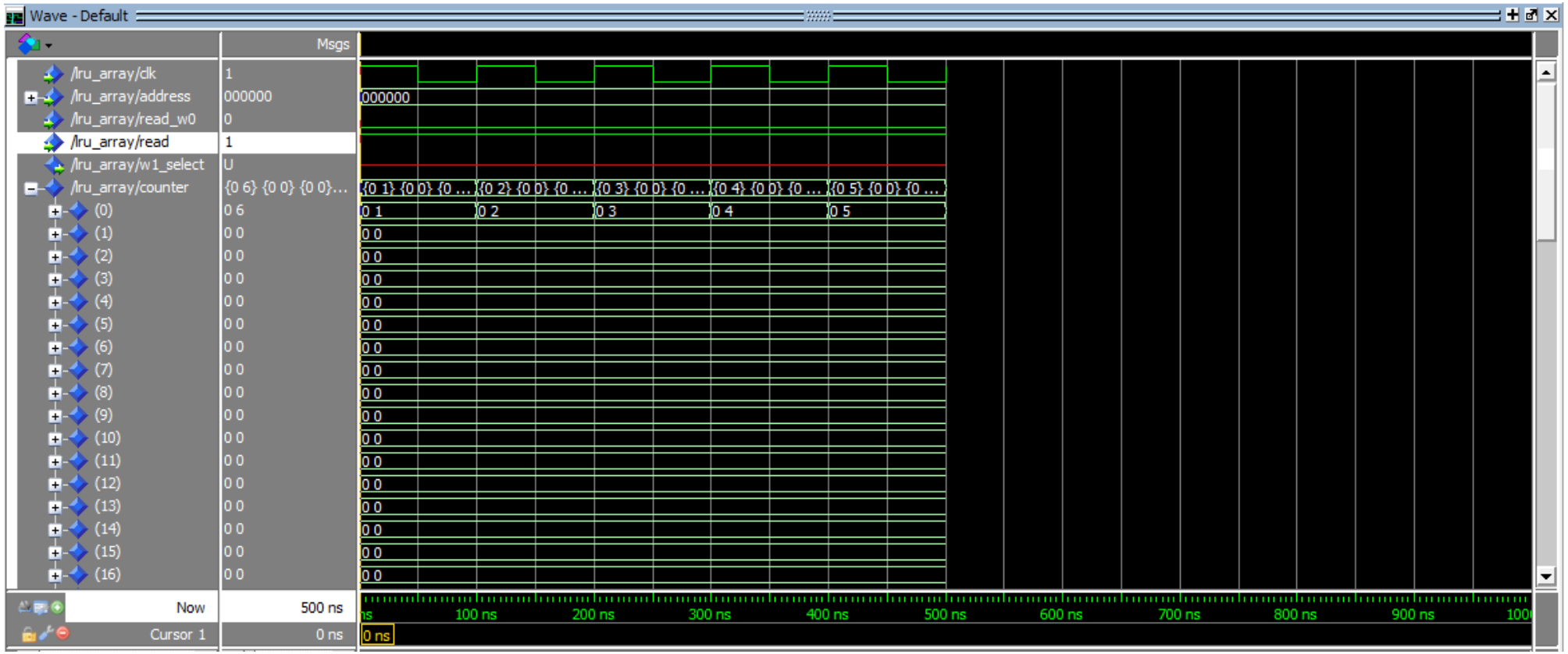
در این حالت miss رخ داده است و علت این است که w0 دارای بیت valid مساوی صفر می باشد و دیگری نیز اصلا تگ برابر با تگ ورودی ندارد:


Wave - Default
Msgs
/miss_hit_logic/tag
/miss_hit_logic/w0
/miss_hit_logic/w1
/miss_hit_logic/hit
/miss_hit_logic/w0_valid
/miss_hit_logic/w1_valid
1111
01111
11010
0
0
0
1111
01111
11010
Now
500 ns
Cursor 1
0 ns
0 ns
100 ns
200 ns
300 ns
400 ns
500 ns
600 ns
700 ns
800 ns
900 ns

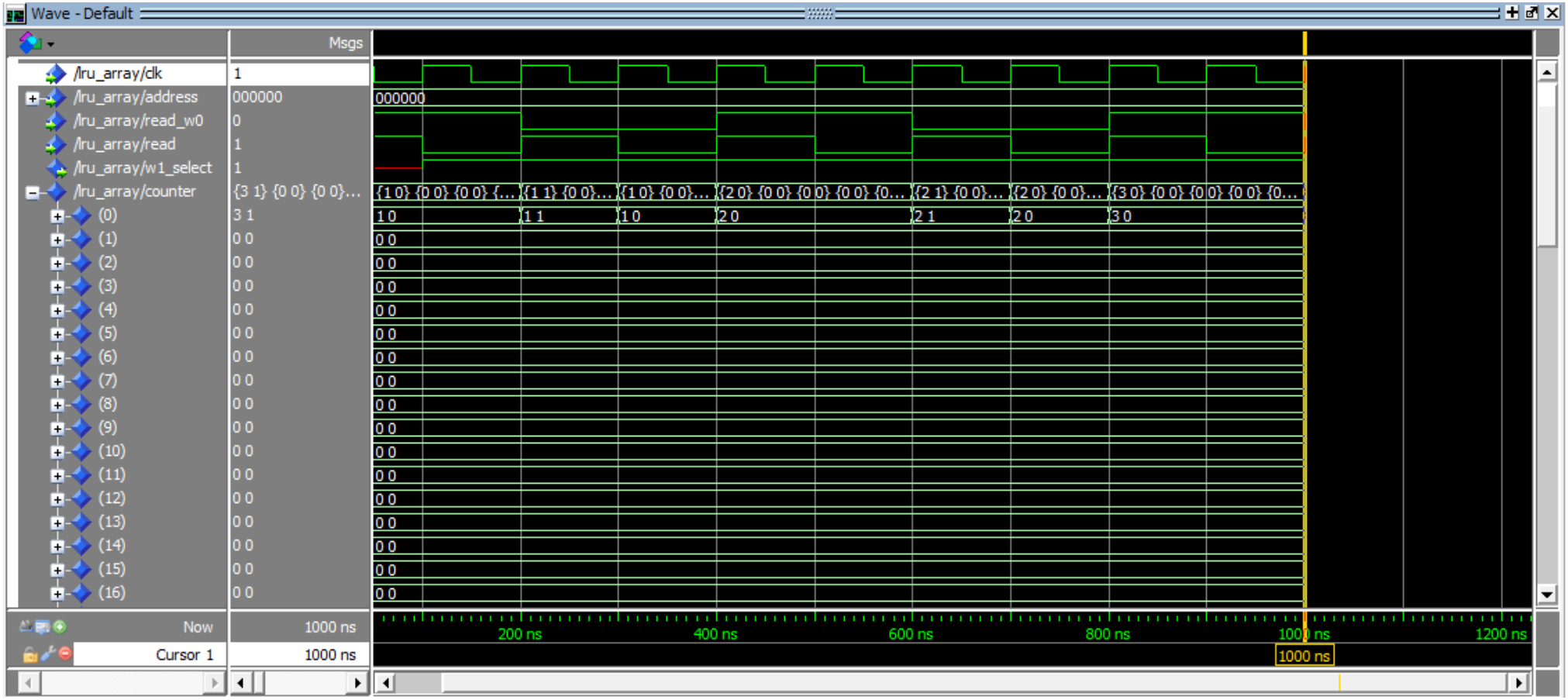
سیاست جایگزینی (lru array) :

این ماژول حساس به کلاک است و دارای سه سیگنال کنترلی read و read_w0 و read_w1است که هر وقت read برابر یک باشد به این معناست که cache در حال خواندن است و شمارنده به ازای آدرس ورودی یکی زیاد می شود که این افزایش برمبنای سیگنال کنترلی دیگری به نام read_w0 و read_w1 است که اگر اولی یک باشد شمارنده مربوط به w0 زیاد میشود و اگر دومی یک باشد شمارنده مربوط به w1 یکی زیاد می شود

در زیر شکل موج مربوط به حالت read_w0 = 0 , read = 1 را مشاهده میکنید :



در شکل زیر شکل موج در حالت کلی تر را برای این ماژول مشاهده میکنید :

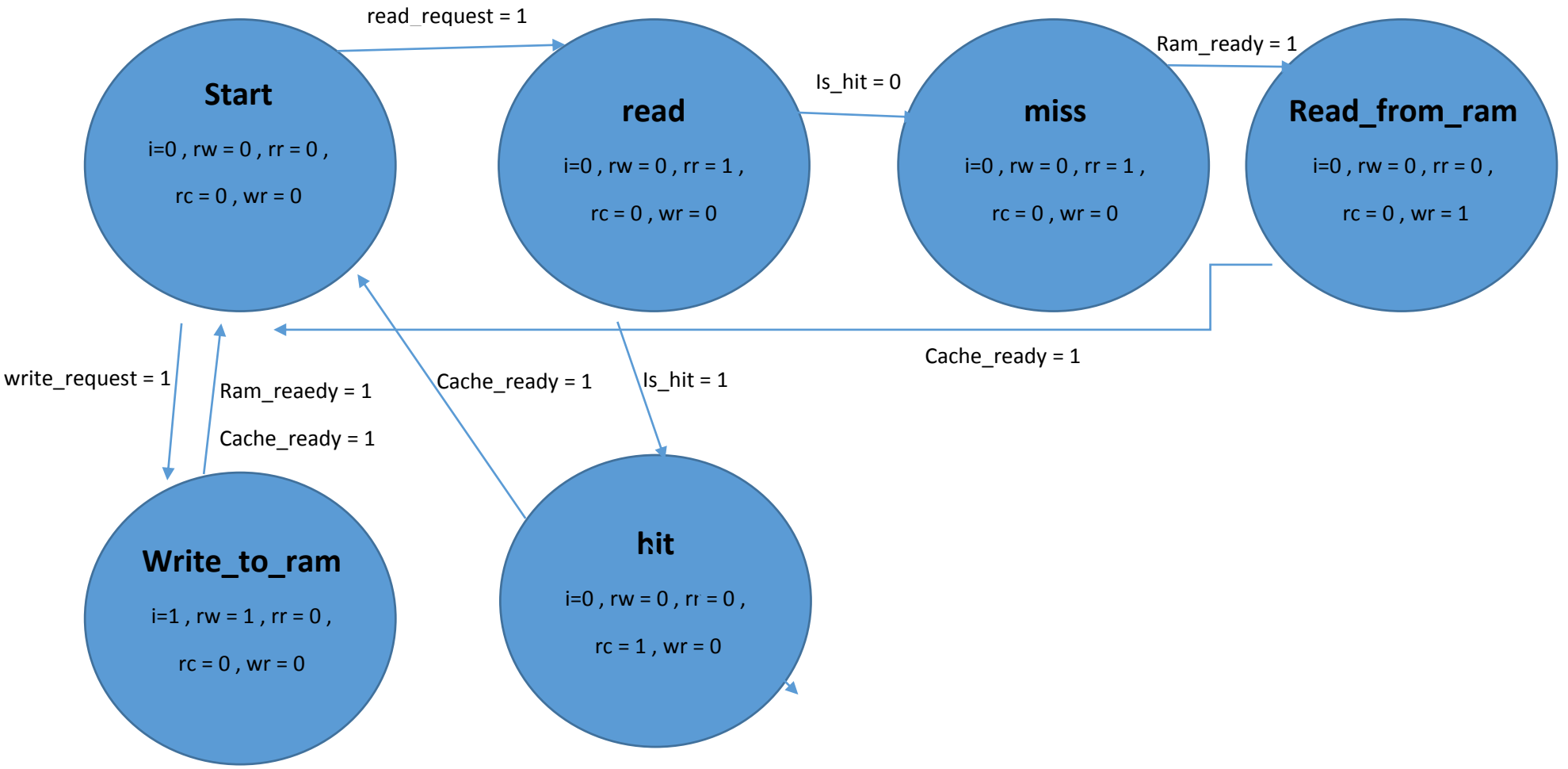


نحوه کارکرد controller :

این ماژول دارای یک سری خروجی هایی ست که تماما سیگنال های کنترلی بقیه ماژول ها هستند که عبارتند از :

invalidate , ram_write ,ram_read ,read_cache,wren برای کارکرد آن یک FSM طراحی کردیم که به صورت زیر است

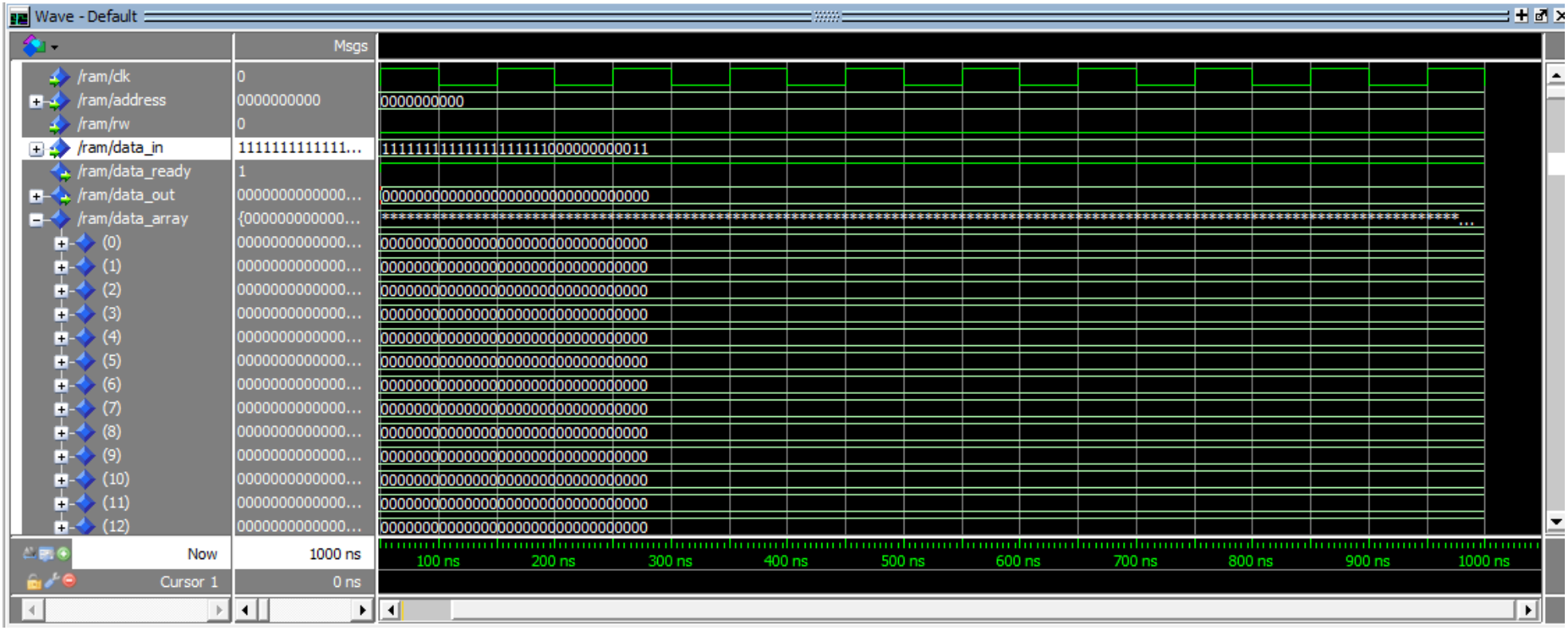
Invalidate : i , ram_write : rw , ram_read : rr, read_cache : rc , wren : wr



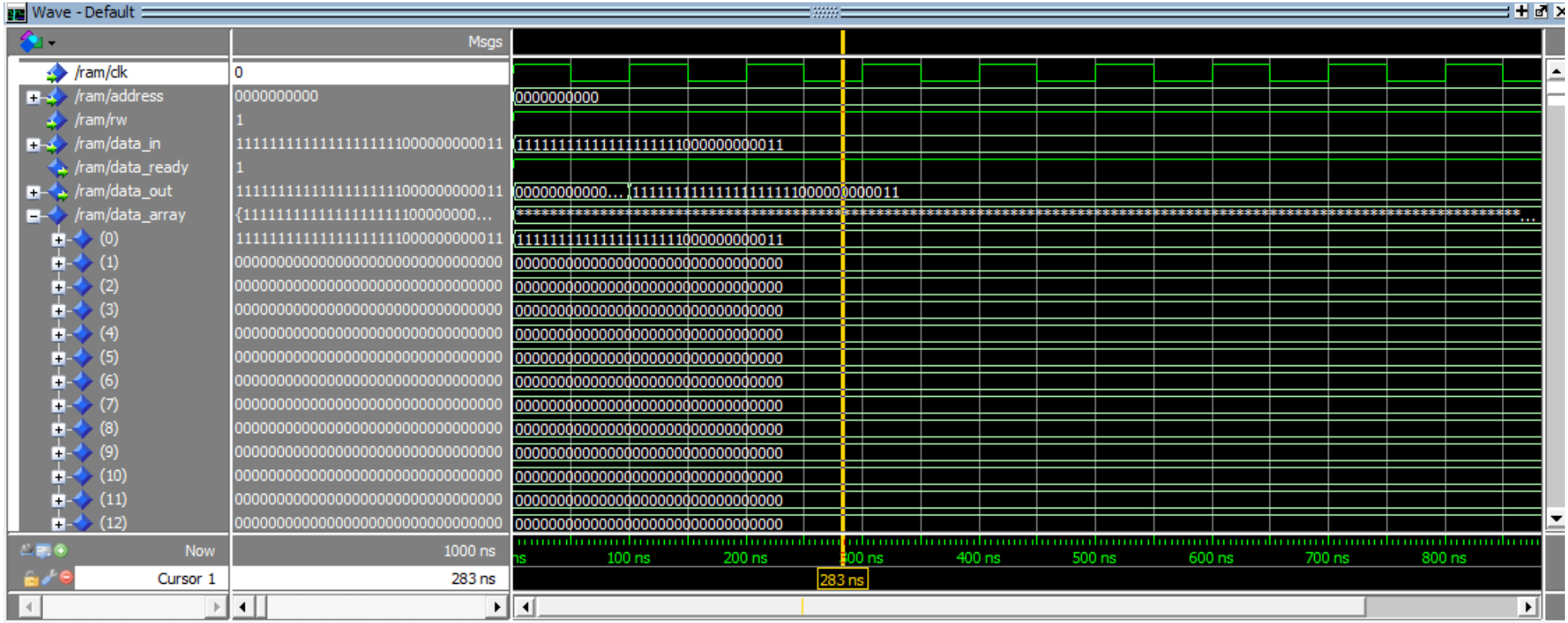
البته توجه کنید که در پیاده سازی FSM بالا برای صرفه جویی در زمان و کم کردن استفاده از زمان کلاک برخی state ها در هم ادغام شده و سیگنال ها بعضا با گذاشتن شروطی در یک state تایین می گردند که البته علت این کار این است که برخی از سیگنال های ورودی این ماژول مانند is_hit چون توسط مدار های ترکیبی تعیین میگردند خیلی زود مشخص می شوند و ربطی به کلاک ندارند

کارکرد ram :

این ماژول همانند data array عمل می کند ولی تنها سیگنال کنترلی آن همان rw است که در زیر شکل موج را به ازای حالت rw = 0 مشاهده میکنید :

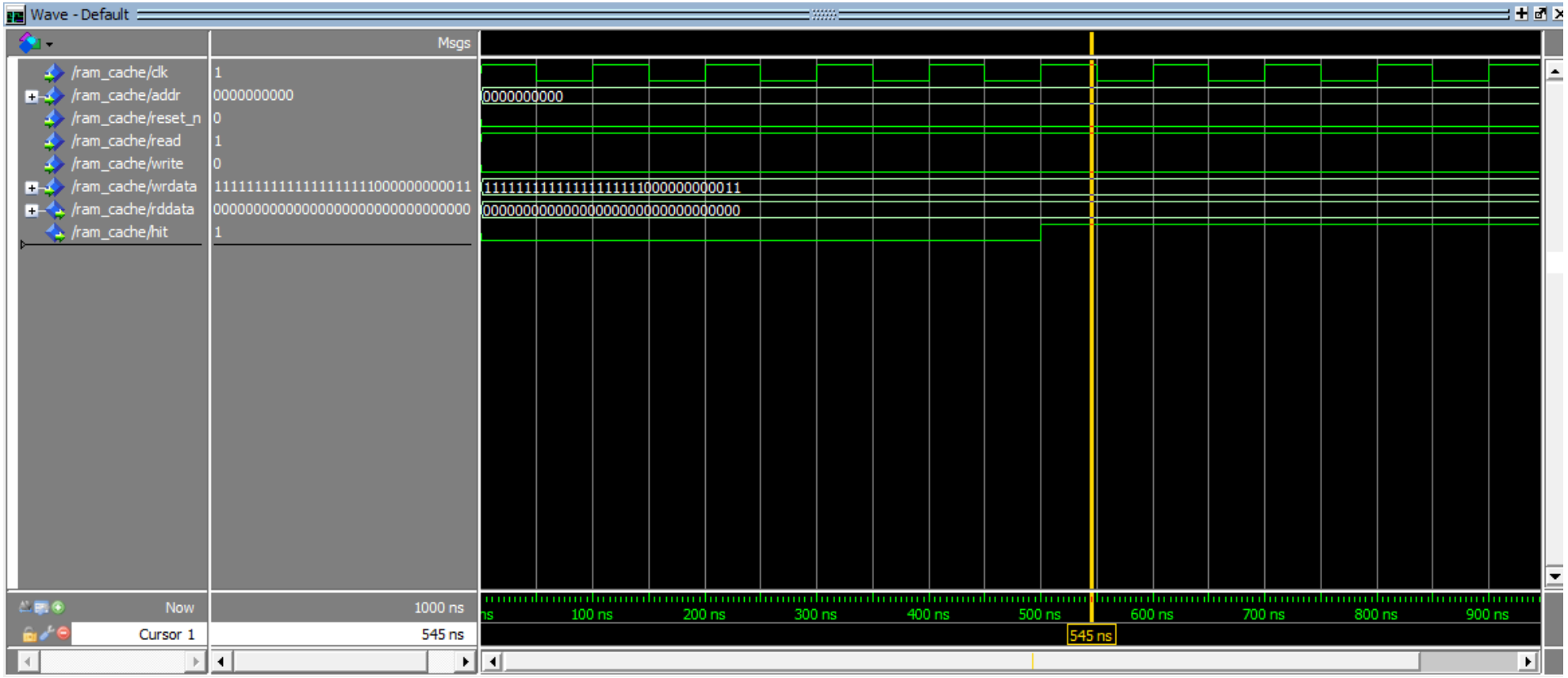


در زیر شکل موج به ازای حالت rw = 1 می بینید که باید دیتا در آدرس مربوطه نوشته شود :

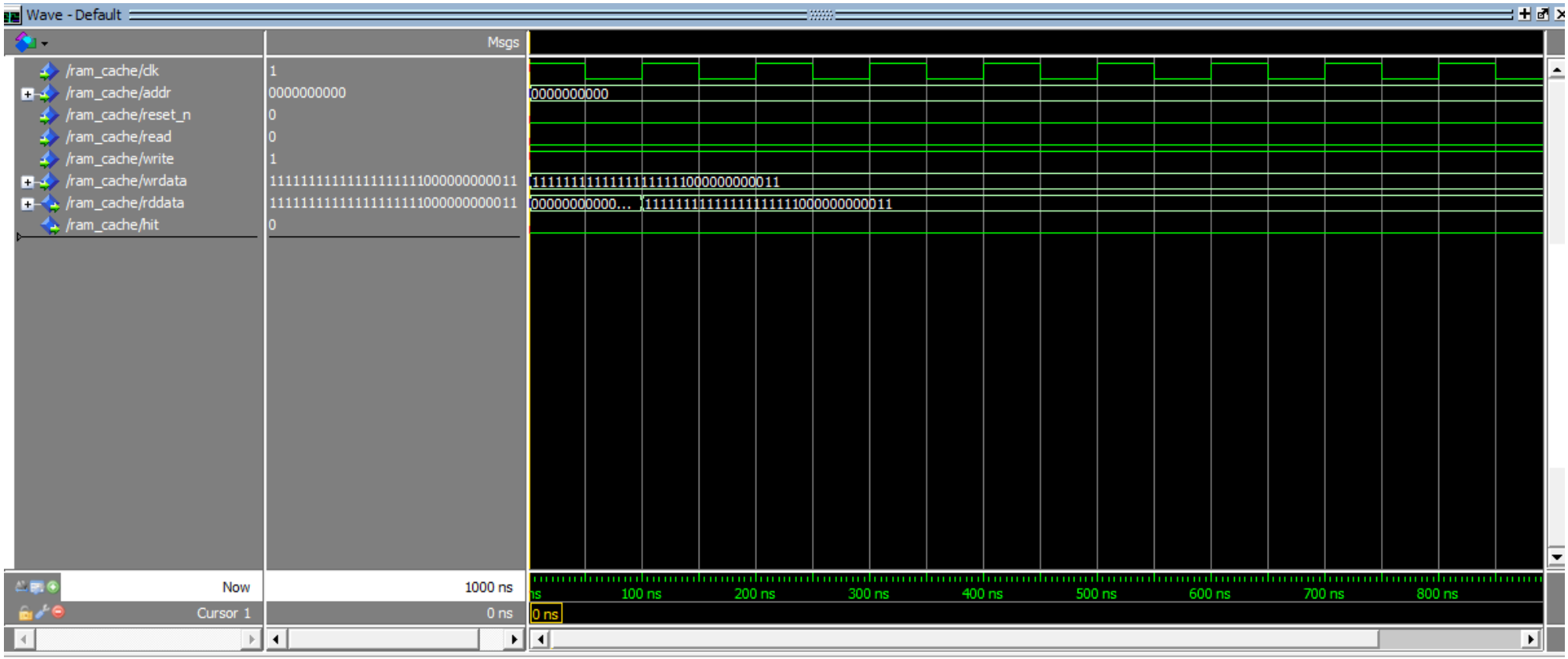


آزمایش ماژول کلی ram_cache :

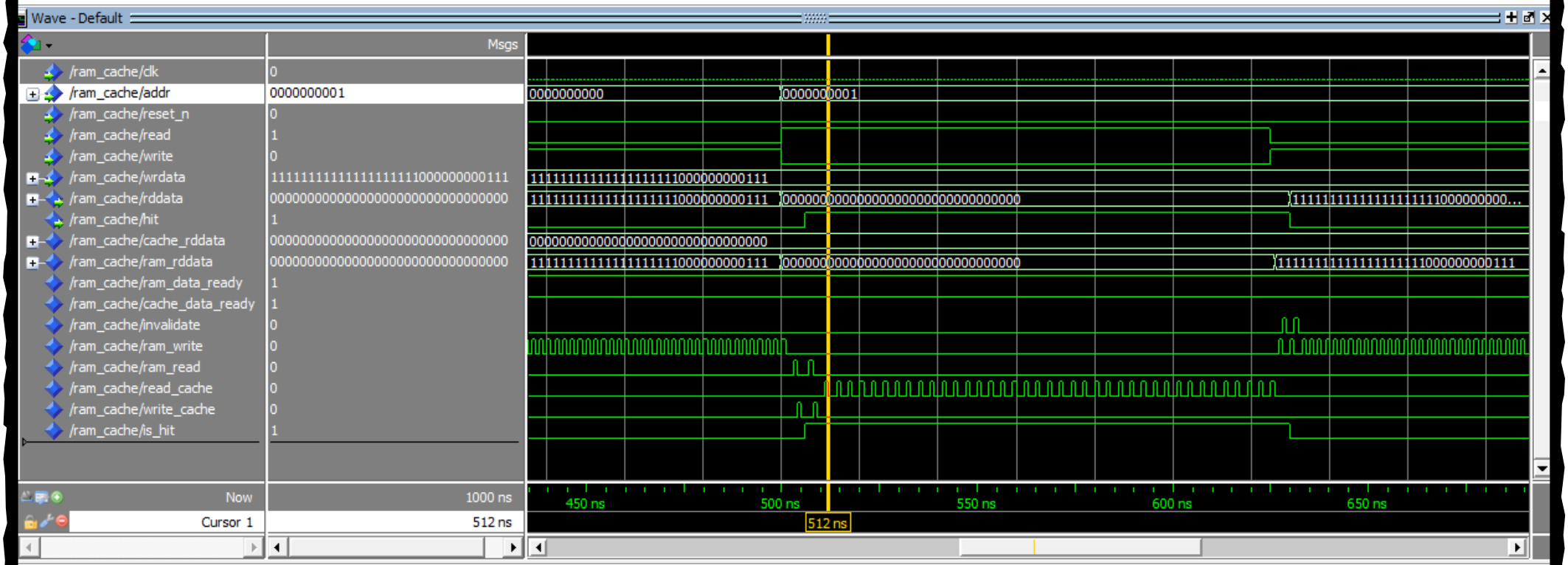
Addr = 0000000000 , reset_n = 0 ,read = 1 , write = 0 , wrdata = 11111111111111111111000000000011



Addr = 0000000000 , reset_n = 0 ,read = 0 , write = 1 , wrdata = 11111111111111111111000000000011



شکل موج حالت کلی ماژول ram_cache :

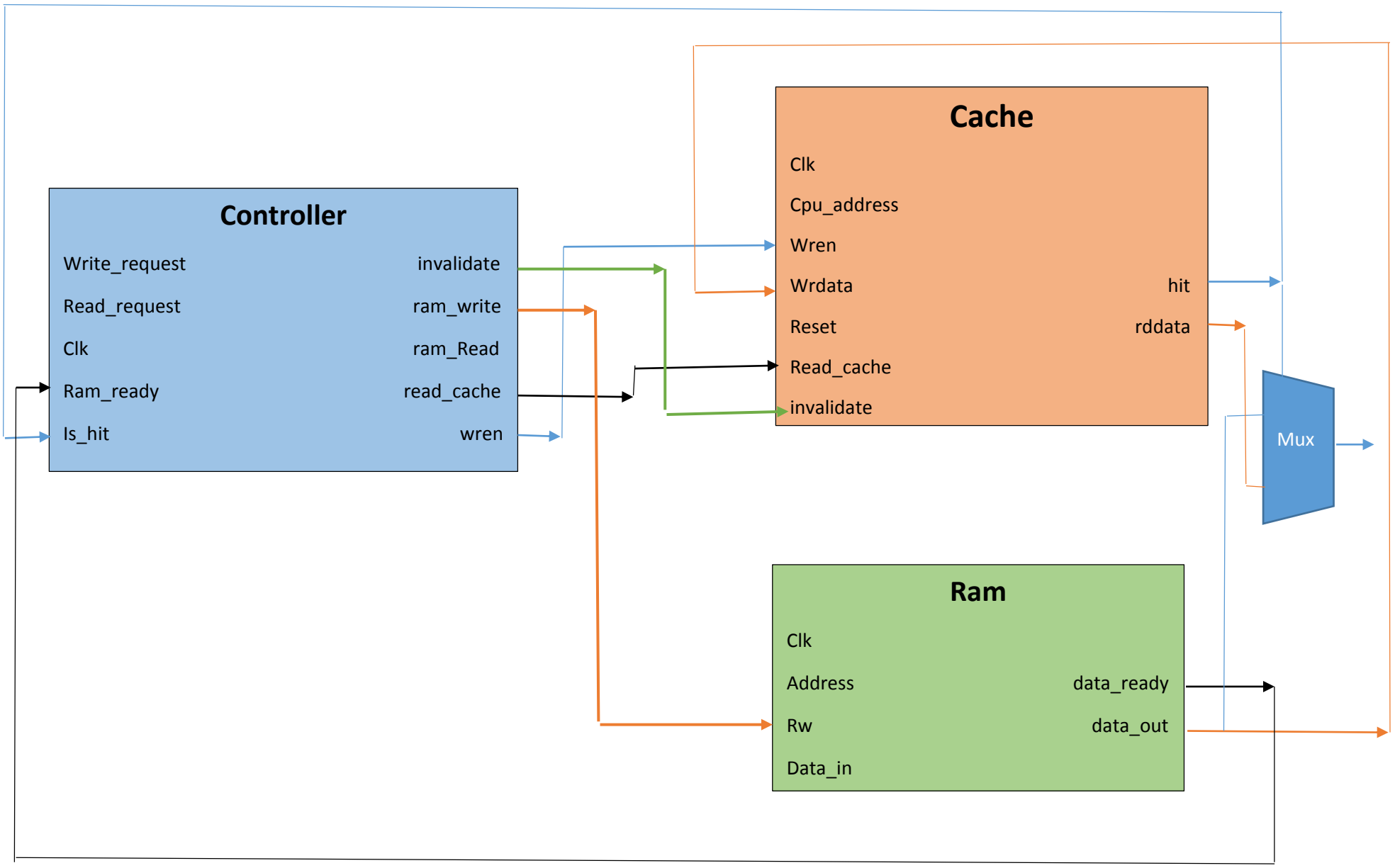



wren
W0_data_array
clk
address
W0_data_array_rddata
W0_wren
wrdata
W1_data_array
clk
address
W1_data_array_rddata
W1_wren
wrdata
Lru
Clk
Address
Read_w0
Read_w1
w1_select
Read
Write
invalidate
W0_tag_valid_array
Clk
Address
W0_wren
w0_tag_valid_output
Reset
Invalidate
wrdata
W1_tag_valid_array
Clk
Address
W1_wren
w1_tag_valid_output
Reset
Invalidate
wrdata
Miss_hit_logic
Tag
hit
W0_tag
w0_valid
W1_tag
w1_valid
نحوه اتصال ماژول ها به یکدیگر
نحوه اتصال ماژول های cache :

توجه : سیگنال های clk , address , invalidate , wrdata , reset جزو ورودی های cache هستند و هر یک مستقیم به ماژول های مربوطه وصل می گردند .

البته در مورد سیگنال address باید گفته که بیت های 0 تا 5 آن وارد data_array ها می شود و بیت 6 تا 9 آن به عنوان wrdata برای tag_validهاست و بیت های 0 تا 5 آن به عنوان address برای tag_valid ها به کار می رود همچنین بیت های 0 تا 5 address بخش lru نیز هستند و بیت های 6 تا 9 به عنوان tag برای miss_hit_logic به کار می روند.

نحوه اتصال ماژول های ram_cache :



توجه : در اینجا سیگنال های clk , address , write_request , read_request , wrdata , reset سیگنال های ورودی هستند.

البته سیگنال wrdataای که از ورودی می آید مستقیما به data_in در رم وصل می شود و wrdata در cacheهمانطور که نشان داده شد از رم سیگنال می گیرد.

خروجی های ماژول نیز خروجی مالتی پلکسر و hit از cache می باشد.