

روش آزمون:

در هر کدام از آزمون های نوشته شده، در ورودی memory تعداد زیادی عدد random generated قرار داده ایم. پس از انجام عملیات تست و دریافت خروجی در مموری، خروجی را با کدی که به صورت high level جواب را بدست می آورد چک میکنیم و در صورت برابری همه خروجی ها با خروجی درست، تست را pass میکنیم.

برای مثال، checker تست اول که حاصل جمع دو بردار را محاسبه میکند، به صورت زیر نوشته شده است:

```
// check correctness
$readmemh("in_memory.txt", in_memory);
$readmemh("out_memory.txt", out_memory);

for(i = 0; i < 16; i = i + 1) begin
    if({out_memory[3 * 16 + i], out_memory[2 * 16 + i]}
        != {{32{in_memory[0 * 16 + i][31]}}, in_memory[0 * 16 + i]} + {{32{in_memory[1 * 16 + i][31]}}, in_memory[1 * 16 + i]}}
        correctness = 0;

    $display("calculated = %h, correct = %h",
        $signed({out_memory[3 * 16 + i], out_memory[2 * 16 + i]}),
        $signed({{32{in_memory[0 * 16 + i][31]}}, in_memory[0 * 16 + i]} + {{32{in_memory[1 * 16 + i][31]}}, in_memory[1 * 16 + i]}));
end

if(correctness == 1)
    $display("test passed!");
else
    $display("test failed!");
```

در ادامه به بررسی آزمون های نوشته شده برای صحت عملکرد پردازنده می پردازیم.

تست اول:

```
$display("test 1: vector addition");
#2 instruction = 9'b00000000; // load A1 mem(0 * 16)
#2 instruction = 9'b00010001; // load A2 mem(1 * 16)
#2 instruction = 9'b10000000; // add
#2 instruction = 9'b01100010; // store A3 mem(2 * 16)
#2 instruction = 9'b01110011; // store A4 mem(3 * 16)
#2
```

در این تست عملیات های load, store و عملیات جمع برداری مورد آزمون قرار گرفته شده است. در ورودی memory مقادیر زیر را برای انجام عملیات جمع ذخیره کرده ایم:

// 0x00000010	// 0x00000000
00000003	00000005
dbbe9427	522c0459
efd1b198	119842e3
76be2ce2	2ece79bf
d7313c26	ff0ba32c
33793846	cea8f677
6597aeae	cd5d5db6
2b2b152a	b6935fc4
309f1f57	4eeb615d
83513051	2219e6dd
fb89fe62	34f10bbf
feecaa7c	fde34993
c6a66aba	9b01b85e
06553e69	11772931
f7e89202	1f2069c7
87a21be4	d02a09ac

پس از اجرای آزمون مقادیر زیر در memory قرار گرفت:

// 0x00000030	// 0x00000020
00000000	00000008
00000000	2dea9880
00000000	0169f47b
00000000	a58ca6a1
ffffffff	d63cdf52
00000000	02222ebd
00000000	32f50c64
ffffffff	e1be74ee
00000000	7f8a80b4
ffffffff	a56b172e
00000000	307b0a21
ffffffff	fccff40f
ffffffff	61a82318
00000000	17cc679a
00000000	1708fbc9
ffffffff	57cc2590

همان طور که در خط اول پیداست حاصل $3 + 5 = 8$ به درستی محاسبه شده است. برای بررسی درستی سایر مقادیر در ماژول آزمون checker نوشته شده است که خروجی آن در ادامه قابل مشاهده است:

```

test 1: vector addition
calculated = 0000000000000008, correct = 0000000000000008
calculated = 000000002dea9880, correct = 000000002dea9880
calculated = 000000000169f47b, correct = 000000000169f47b
calculated = 00000000a58ca6a1, correct = 00000000a58ca6a1
calculated = ffffffff63cdf52, correct = ffffffff63cdf52
calculated = 000000000222ebd, correct = 000000000222ebd
calculated = 0000000032f50c64, correct = 0000000032f50c64
calculated = ffffffff1be74ee, correct = ffffffff1be74ee
calculated = 000000007f8a80b4, correct = 000000007f8a80b4
calculated = ffffffff56b172e, correct = ffffffff56b172e
calculated = 00000000307b0a21, correct = 00000000307b0a21
calculated = ffffffffccff40f, correct = ffffffffccff40f
calculated = ffffffff61a82318, correct = ffffffff61a82318
calculated = 0000000017cc679a, correct = 0000000017cc679a
calculated = 000000001708fbc9, correct = 000000001708fbc9
calculated = ffffffff57cc2590, correct = ffffffff57cc2590
test passed!

```

تست دوم:

```

$display("test 2: vector multiplication");
#2 instruction = 9'b000000000; // load A1 mem(0 * 16)
#2 instruction = 9'b000100001; // load A2 mem(1 * 16)
#2 instruction = 9'b110000000; // multiply
#2 instruction = 9'b011000010; // store A3 mem(2 * 16)
#2 instruction = 9'b011100011; // store A4 mem(3 * 16)
#2

```

در این تست عملیات های `load`, `store` و عملیات ضرب برداری مورد آزمون قرار گرفته شده است. در ورودی `memory` مقادیر زیر را برای انجام عملیات ضرب ذخیره کرده ایم:

// 0x00000010	// 0x00000000
00000003	00000005
dbbe9427	522c0459
efd1b198	119842e3
76be2ce2	2ece79bf
d7313c26	ff0ba32c
33793846	cea8f677
6597aeae	cd5d5db6
2b2b152a	b6935fc4
309f1f57	4eeb615d
83513051	2219e6dd
fb89fe62	34f10bbf
feecaa7c	fde34993
c6a66aba	9b01b85e
06553e69	11772931
f7e89202	1f2069c7
87a21be4	d02a09ac

پس از اجرای آزمون مقادیر زیر در memory قرار گرفت:

```
// 0x00000030 // 0x00000020
00000000      0000000f
f45ccf98      54461d8f
fee34d12      40f9a9c8
15b5f191      34f64e9e
0026f3de      c5648888
f6144c1d      ab5c6c8a
ebe7d57e      1f8165b4
f39e65c5      0cf4ca28
0efd301c      3b8b599b
ef642ae7      c8687bed
ff13d45f      fc25011e
0002458c      96774134
169ff53d      968ae04c
006e9bc9      d91ec319
ff042368      97ec518e
167dd6d1      a74dc130
```

همان طور که در خط اول پیداست حاصل $3 * 5 = 15$ به درستی محاسبه شده است. برای بررسی درستی سایر مقادیر در ماژول آزمون checker نوشته شده است که خروجی آن در ادامه قابل مشاهده است:

```
test 2: vector multiplication
calculated = 000000000000000f, correct = 000000000000000f
calculated = f45ccf9854461d8f, correct = f45ccf9854461d8f
calculated = fee34d1240f9a9c8, correct = fee34d1240f9a9c8
calculated = 15b5f19134f64e9e, correct = 15b5f19134f64e9e
calculated = 0026f3dec5648888, correct = 0026f3dec5648888
calculated = f6144c1dab5c6c8a, correct = f6144c1dab5c6c8a
calculated = ebe7d57e1f8165b4, correct = ebe7d57e1f8165b4
calculated = f39e65c50cf4ca28, correct = f39e65c50cf4ca28
calculated = 0efd301c3b8b599b, correct = 0efd301c3b8b599b
calculated = ef642ae7c8687bed, correct = ef642ae7c8687bed
calculated = ff13d45ffc25011e, correct = ff13d45ffc25011e
calculated = 0002458c96774134, correct = 0002458c96774134
calculated = 169ff53d968ae04c, correct = 169ff53d968ae04c
calculated = 006e9bc9d91ec319, correct = 006e9bc9d91ec319
calculated = ff04236897ec518e, correct = ff04236897ec518e
calculated = 167dd6d1a74dc130, correct = 167dd6d1a74dc130
test passed!
```