

زاگرس

مواصفات زاگرس - تجدید نظر شده در فروردین ۱۴۰۱.

معرفی

زاگرس یک سامانه محاسبات مجازی مختص سیستم‌های نهفته است. هدف این ماشین مجازی ارائه‌ی قابلیت بروزرسانی زنده با بالاترین چگالی کد است. فواید ثانویه طراحی آن، پوسته (shell) ای واکنشی (reactive)، قابل برنامه‌نویسی و همچنین بررسی بلادرنگ است. زاگرس مختص سامانه‌های روی تراشه طراحی شده است به شرطی که حداقل به اندازه‌ی ESP8266 منابع داشته باشند. (این سامانه‌ی روی تراشه در بازارهای غربی قیمت عمده ۴ دلار و در بازارهای چینی قیمت عمده ۲ دلار دارد).

این مواصفات در نظر دارد تا پایه‌ای برای ساخت سامانه‌های زاگرس فراهم کند. انتظار می‌رود پیاده‌کنندگان دستگاه‌های I/O آن را برای سخت‌افزار مورد نظر خود گسترش دهند یا اصلاح کنند.

بررسی اجمالی

- زاگرس سامانه ای ۳۲ بیتی است
- پشته‌ی داده آن ۳۲ عضو در خود جای می‌دهد
- پشته‌ی آدرس آن ۱۲۸ عضو در خود جای می‌دهد
- همه‌ی عناصر پشته و ثبات‌های داخلی آن ۳۲ بیتی هستند
- هشت واحد پردازش مرکزی مجازی در آن با زمان‌بندی نوبت گردشی دایر هستند
- هر هسته ۲۴ کلمه حافظه داخلی، یک پشته‌ی داده، یک پشته‌ی آدرس، و ثبات‌های اشاره‌گر دستورالعمل (IP)، اشاره‌گر پشته (SP) و اشاره‌گر بازگشت (RP) را دارند.
- هسته‌ها می‌توانند فعال و غیر فعال شوند
- پیاده‌سازی مرجع، دسترسی به ۶۴ کیلو بایت حافظه تصادفی را فراهم می‌کند

حافظه

حافظه این سامانه از یک فضای آدرس تشکیل می‌شود. آدرس‌ها به صورت متوالی تعیین شده‌اند.

آدرس دهی

حافظه می‌تواند با اندازه‌های بایت، نیم‌کلمه و کلمه آدرس‌دهی شود.

تراز

کلمه‌ها باید با کرانه ۴ بایتی تراز شوند. نیازی نیست نیم کلمه‌ها و بایت‌ها هنگامی که به عنوان داده استفاده می‌شوند تراز شوند. برای خواندن یک بایت ۸ بیت اول مقدار نگاشته‌ی آدرس را می‌خوانیم و می‌نویسیم و باقی بیت‌ها را نادیده می‌گیریم. برای نمونه:

Address	0-7	8-15	16-23	24-31
0000	byte	half		word
0001	byte			
0010	byte	half		
0011	byte			
0100	byte	half		word
0101	byte			
0110	byte	half		
0111	byte			

نگاشت حافظه

نخستین ۱۹۲ بایت حافظه برای محوطه واسطه‌های I/O رزرو شده اند. از اینجا به بعد حافظه عموماً در اختیار برنامه‌ها قرار دارد. با این حال بعضی دستگاه‌ها می‌توانند بخشی از فضای حافظه را برای I/O حافظه نگاشتی تخصیص دهند.

دستگاه‌ها می‌توانند مناطق حافظه بیشتری نگاشت کنند که باید در پیاده سازی مستند و مطالعه شود.

دستورالعمل‌ها و پردازش

زاگرس دستورالعمل‌ها را به عنوان بسته‌های کلمه اندازه پردازش می‌کند (اگر کلمه‌ها ۳۲ بیتی باشند، دستورالعمل‌ها ۳۲ بیت واکشی می‌شوند). هر بسته می‌تواند تا چهار جایگاه دستورالعمل و داده‌ی مربوط داشته باشد. دستورالعمل‌ها و اطلاعات آن باید در یک بسته جا شوند.

بسته‌های دستورالعمل باید در کرانه‌های کلمه‌ای تراز شوند. برای مثال می‌توانیم دنباله‌ای از بسته‌ها به شکل زیر داشته باشیم:

```
lb 1
ad
re

du
mu
re
..
```

به شرط اینکه مولفه‌های خطی (inline) در یک کلمه باشند. کد زیر درست خواهد بود:

```
du
mu
lb 45
```

در حالی که کد زیر خیر:

```
du
mu
lh 45
```

دستور lh (بخوانیم load half) یک نیم کلمه را بارگذاری می‌کند پس مولفه خطی آن یک مقدار ۱۶ بیتی در نظر گرفته می‌شود. که مجموعه این سه خط دستور را ۴۰ بیت می‌کند که از ۳۲ بیت کلمه بیشتر است.

موارد خاص

- دستور lw مقدار کلمه موجود در بسته بعدی را در پشته push می‌کند.
- دستور lh مقدار نیم کلمه (۱۶ بیت) موجود در جایگاه دستورالعمل بعدی را در پشته push می‌کند.
- دستور lb مقدار بایت (۸ بیت) موجود در جایگاه دستورالعمل بعدی را در پشته push می‌کند.
- دستورهای cr، re، cj، ju، cc، ca و ti اشاره گر دستورالعمل (IP) را تغییر می‌دهند و نمی‌توانند توسط هیچ دستورالعملی به جز no دنبال شوند. هر دستورالعملی که بعد از آن‌ها خطی شود توسط زاگرس به عنوان no تلقی می‌شود.

دستورالعمل‌ها در یک کلمه ۳۲ بیتی بسته‌بندی می‌شوند و به چهار جایگاه ۸ بیتی تقسیم می‌شوند.

دستور lh می‌تواند در جایگاه ۰ و ۱ استفاده شود. چون به دنبال آن مقداری ۱۶ بیتی (۲ جایگاهی) می‌آید که نباید از کران یک کلمه ی بسته ی دستورالعملش خارج شود. در غیر این صورت مولفه ی خطی (داده ی) بیرون از بسته دستورالعمل قرار می‌گیرد. lb می‌تواند در جایگاه‌های ۰، ۱ و ۲ قرار گیرد چون به دنبال آن یک مقدار ۸ بیتی می‌آید.

هر دستورالعملی که به دنبال دستورالعمل‌هایی بیاید که اشاره گر دستورالعمل (IP) را تغییر می‌دهند نادیده گرفته می‌شود (یا به عنوان no تلقی می‌شوند). این دستورالعمل‌ها cr، re، cj، ju، cc، ca و ti هستند.

اشاره گر دستورالعمل (IP) بعد از پردازش یافتن بسته ۱ واحد افزایش می‌یابد.

فهرست دستورالعمل‌ها

پشته ی (داده) با ds و پشته ی آدرس با as نشان داده شده.

#n	xn	کد	وضعیت پشته‌ها	شرح	تلفظ انگلیسی								
00	00	<table><tr><td>..</td><td>..</td><td>..</td><td>..</td></tr></table>		عملیات خالی	nop				
..										
01	01	<table><tr><td>lw</td><td>..</td><td>..</td><td>..</td></tr><tr><td>nn</td><td>nn</td><td>nn</td><td>nn</td></tr></table>	lw	nn	nn	nn	nn	ds.push	مقدار ۳۲ بیتی nnnnnnnn را در پشته push می‌کند.	load word
lw										
nn	nn	nn	nn										
02	02	<table><tr><td>lh</td><td>nn</td><td>nn</td><td>..</td></tr></table>	lh	nn	nn	..	ds.push	مقدار ۱۶ بیتی nnnn را در پشته push می‌کند.	load half				
lh	nn	nn	..										
03	03	<table><tr><td>lb</td><td>nn</td><td>..</td><td>..</td></tr></table>	lb	nn	ds.push	مقدار ۸ بیتی nn را در پشته push می‌کند.	load byte				
lb	nn										
04	04	<table><tr><td>fw</td><td>..</td><td>..</td><td>..</td></tr></table>	fw	a=ds.pop ds.push i	واکشی مقدار ۳۲ بیت	fetch word				
fw										
05	05	<table><tr><td>fh</td><td>..</td><td>..</td><td>..</td></tr></table>	fh	a=ds.pop ds.push i	واکشی مقدار ۱۶ بیت	fetch half				
fh										
06	06	<table><tr><td>fb</td><td>..</td><td>..</td><td>..</td></tr></table>	fb	a=ds.pop ds.push i	واکشی مقدار ۸ بیت	fetch byte				
fb										
07	07	<table><tr><td>sw</td><td>..</td><td>..</td><td>..</td></tr></table>	sw	i, a=ds.pop 2	ذخیرسازی مقدار ۳۲ بیت	store word				
sw										
08	08	<table><tr><td>sh</td><td>..</td><td>..</td><td>..</td></tr></table>	sh	i, a=ds.pop	ذخیرسازی مقدار ۱۶ بیت	store half				
sh										
09	09	<table><tr><td>sb</td><td>..</td><td>..</td><td>..</td></tr></table>	sb	i=ds.pop a=ds.pop	ذخیرسازی مقدار ۸ بیت	store byte				
sb										

10	0A	du	i=ds.pop ds.push i, i	تکثیر مقدار بالای پشته	dupe
11	0B	dr	ds.pop	رها کردن مقدار بالای پشته	drop
12	0C	sp	i, j=ds.pop 2 ds.push i, j	تعویض دو مقدار بالاش پشته	swap
13	0D	pu	ds.pop as.push	بردن مقدار بالای پشته به پشته‌ی آدرس	push address
14	0E	po	as.pop ds.push	بردن مقدار بالای پشته‌ی آدرس به پشته‌ی داده	pop address
15	0F	eq	ds.pop 2 ds.push	بررسی برابری	equal
16	10	ne	ds.pop 2 ds.push	بررسی نابرابری	not equal
17	11	lt	ds.pop 2 ds.push	بررسی کوچکتري	less than
18	12	gt	ds.pop 2 ds.push	بررسی بزرگتری	greater than
19	13	ad	ds.pop 2 ds.push	جمع	add

20	14	su	ds.pop 2 ds.push	تفریق	sub
21	15	mu	ds.pop 2 ds.push	ضرب	multiply
22	16	dm	ds.pop 2 ds.push r, d	تقسیم و باقی‌مانده	divide and reminder
23	17	md	ds.pop x3 ds.push r, d	ضرب، تقسیم و باقی‌مانده	multiply, divide and reminder
24	18	an	ds.pop 2 ds.push	و باینری	and
25	19	or	ds.pop 2 ds.push	یا باینری	or
26	1A	xo	ds.pop 2 ds.push	یا انحصاری باینری	xor
27	1B	nt	ds.pop ds.push	نفی باینری	not
28	1C	sl	i, j=ds.pop 2 ds.push j	شیفت چپ	shift left
29	1D	sr	i, j=ds.pop 2 ds.push j	شیفت راست	shift right
30	1E		ds.pop 4	بسته کردن ۴ بایت به یک مقدار ۳۲	pack

		pa	ds.push	بیتی	
31	1F	un	ds.pop ds.push 4	باز کردن ۱ مقدار ۳۲ بیتی به چهار بایت	unpack
32	20	rl		نسبی در نظر گرفتن پرش یا فراخوانی بعدی	relative
33	21	ca - - -	ds.pop as.push	فراخوانی	call
34	22	cc - - -	c, a=ds.pop 2 as.push?	فراخوانی شرطی	conditional call
35	23	ju - - -	ds.pop	پرش	jump
36	24	cj - - -	c, a=ds.pop 2	پرش شرطی	conditional jump
37	25	re - - -	as.pop	بازگشت	return
38	26	cr - - -	ds.pop as.pop	بازگشت شرطی	conditional return
39	27	sv	a, i=ds.pop 2	تنظیم بردار وقفه	set interrupt vector
40	28			ایست وقفه	halt interrupts

		hi			
41	29	si		آغاز وقفه	start interrupts
42	2A	ti	ds.pop	راه اندازی وقفه	trigger interrupt
43	2B	ii	ds.pop	فعل و انفعال I/O	i/o interaction
44	2C	hs		ایست سامانه	halt system
45	2D	ic	a, i=ds.pop 2	راه اندازی هسته	init core
46	2E	ac	ds.pop	فعال سازی هسته	active core
47	2F	pc	ds.pop	مکث (موقت) هسته	pause core
48	30	sc		تعطیل هسته ی فعلی	suspend current core
49	31	rr	ds.pop ds.push	خواندن ثبات هسته فعلی	read register
50	32		a, i=ds.pop 2	نوشتن در ثبات هسته فعلی	write register

		<table><tr><td>wr</td><td>..</td><td>..</td><td>..</td></tr></table>	wr			
wr						
51	33	<table><tr><td>cp</td><td>..</td><td>..</td><td>..</td></tr></table>	cp	ds.pop 3	کپی کردن n کلمه از آدرس a1 تا a2	copy
cp						
52	34	<table><tr><td>bc</td><td>..</td><td>..</td><td>..</td></tr></table>	bc	ds.pop 3 ds.push	مقایسه n کلمه از آدرس a1 تا a2	block compare
bc						
53	35	<table><tr><td>uu</td><td>..</td><td>..</td><td>..</td></tr></table>	uu		بی علامت در نظر گرفتن عملیات بعدی	unsigned
uu						

دستگاه ها