

طراحی زبان توصیف سخت افزار AHDSL

کریم فائز
دانشیار دانشکده مهندسی برق

مسعود خسروی
دانشجوی کارشناسی ارشد دانشکده مهندسی کامپیوتر

دانشگاه صنعتی امیرکبیر

چکیده

در این مقاله زبان توصیف سخت افزار AHDSL (*Amirkabir Hardware Description and Synthesising Language*) مورد بررسی قرار می‌گیرد. این زبان پس از بررسی زبانهای متداول توصیف سخت افزاری موجود، تعریف و مترجم آن تهیه گردیده است. این زبان بگونه‌ای طراحی شده که می‌تواند انواع مدارهای منطقی را با ساده‌ترین راه ممکن توصیف نماید. در این مقاله ابتدا به بررسی این زبان پرداخته و سپس نحوه توصیف پردازنده همه منظوره PARWAN که برای مقاصد آموزشی طراحی شده است مورد بررسی قرار می‌گیرد.

مقایسه نتایج حاصل از این زبان با کارهای مشابه‌ای که به زبان VHDL انجام شده است نشان می‌دهد که توصیف AHDSL در عین سادگی، سخت‌افزاری مشابه توصیف VHDL را ارائه می‌دهد. همچنین زمان ترجمه آن بسیار کوتاهتر از توصیف با VHDL است.

Design of AHDSL Hardware Description Language

M. Khosravi

Graduate Student Computer, Eng. Dept.

K. Faez

Associate Prof. Elect. Eng. Dept.

Amirkabir Univ. of Tech.

ABSTRACT

In this paper, we study the AHDSL (Amirkabir Hardware Description and Synthesising Language) hardware description language. By considering other hardware description languages, this language was defined and a compiler is designed for it. This language will provide a simple tool to describe any digital circuit. At this paper, primarily we describe the AHDSL, and then the description of PARWAN educational general purpose processor using AHDSL is considered. Finally the comparison between AHDSL and VHDL description of PARWAN is demonstrated. This comparison shows that hardware is similar to the corresponding VHDL results. In addition, the compiling time of AHDSL description is less than that of the VHDL description.

یک چیپ صرف می‌شد، گاهی به اندازه زمانی بود که چیپ طراحی شده می‌توانست در بازار، فروش داشته باشد و این امر باعث یک تناقض در پیشرفت پروسه طراحی بود. زیرا نیاز به مدارهای پیچیده روز بروز افزایش می‌یافت و ساخت مدارهای

۱- مقدمه

با شروع دهه ۸۰ بحران در تکنولوژی ساخت مدارهای مجتمع آغاز شد، زیرا پیچیدگی مدارهای مجتمع بگونه‌ای بالا رفته بود که تا آن زمان سابقه نداشت و زمانی که برای طراحی

شده است.

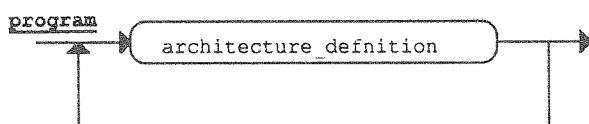
۲- زبان AHDSL

پس از بررسیهای لازم، قواعدی که در ذیل به توضیح آنها خواهیم پرداخت برای زبان AHDSL انتخاب و یک مترجم برای آن تهیه گردیده که می تواند برنامه هایی را که به این زبان توصیف گردیده اند به یک فهرست در یچه (gate list) تبدیل نماید.

۱-۲- ساختار یک برنامه AHDSL

در AHDSL هر برنامه می تواند شامل تعدادی معماری باشد و هر کدام از این معماریها می توانند یک شیء را توصیف نمایند. البته در قواعدی که در حال حاضر تهیه شده هر کدام از معماریها بصورت مجزا سنتز می گردند و امکان اینکه یک معماری از یک معماری دیگر استفاده کند وجود ندارد. شکل (۱) ساختار برنامه های AHDSL را نشان می دهد.

شکل (۲) برنامه ای شامل دو معماری به نامهای A1 و A2 را نشان می دهد.



شکل (۱) نمودار ساختار یک برنامه در AHDSL

--test example for a AHDSL programs

architecture A1()

```
{
-- A1 Body.....
}
```

architecture A2()

```
{
-- A2 Body.....
}
```

شکل (۲) یک برنامه شامل دو معماری A1 و A2 به زبان AHDSL

۲-۲- قواعد تعریف یک معماری

هر معماری شامل دو قسمت است. قسمت اول معرفی معماری است، که شامل نام و تعریف ورودیها و خروجیها است،

پیچیده نیز زمان زیادتری لازم داشت. به همین دلیل نیاز به نرم افزارهایی که بتوانند طراحی را بصورت اتوماتیک انجام دهند روز بروز بیشتر احساس می شد. به همین دلیل روشهای مختلف طراحی اتوماتیک ابداع گردید. روش طراحی اتوماتیک دارای مزایای عمده ذیل است:

الف - قابلیت استفاده (useability)

طراحان می توانند در سطوح بالاتری با مسأله برخورد نمایند و نیازی به دانستن جزئیات طراحی یک چیپ نداشته باشند، به همین دلیل این ابزارها می توانند باعث توسعه دید طراحان شده و موجب تسهیل در ساخت مدارهای پیچیده تر می گردند [۱].

ب - کیفیت طراحی

امکان "تصحیح به کمک طراحی" (correct by design) این مزیت را دارد که طرح می تواند عاری از خطاهای انسانی باشد. زیرا خطاهایی که در توصیف سطح بالا وجود دارد به آسانی قابل تشخیص و پیگیری است [۱].

ج - کمیت تولید

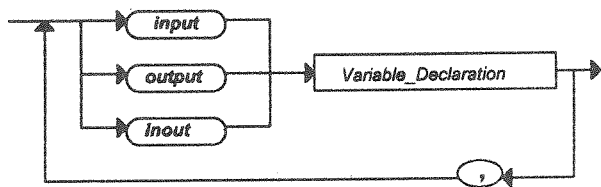
به دلیل اینکه طراحان با جزئیات درگیر نمی گردند، زمان طراحی بسیار تقلیل می یابد که این خود باعث می گردد که بتوان مدارهای بیشتری را طراحی و به بازار ارائه نمود [۱].

البته هر روش دارای معایبی نیز هست و ایراد طراحی اتوماتیک در این است که ابزارهای مورد استفاده باید در محدوده ای به وسعت تمامی طرحهایی که ممکن است به ذهن انسان برسد، قابلیت استفاده داشته باشد و به همین دلیل مدارهای تولید شده، به میزان طرح انجام شده توسط یک انسان که در یک کاربرد خاص مهارت کسب نموده است، جمع و جور نخواهد بود [۲].

در این مقاله در ابتدا زبان AHDSL مورد بررسی قرار می گیرد [۴]. این زبان پس از بررسی زبانهای موجود توصیف سخت افزار، طراحی و یک مترجم برای آن تهیه گردیده است. این زبان برخلاف زبانهای موجود دارای ساختار بسیار ساده ای است و به همین دلیل مترجم این زبان دارای سرعت بسیار خوبی است. AHDSL همانند زبانهای دیگر توصیف سخت افزار یک زبان همروند (Concurrent) است.

پس از بررسی AHDSL، طراحی یک پردازنده همه منظوره مورد بررسی قرار می گیرد. در انتها نیز نتایج حاصل از توصیف پردازنده با زبان VHDL و AHDSL با هم مقایسه شده است. در این مقاله لغتهای کلیدی AHDSL بصورت متمایل نشان داده

OPTIONAL PARAMETER



شکل (۵) قاعده تعریف پارامترهای یک معماری در AHDSL

**architecture Adder4Bit(input a[4],input b[4],
input ci, output co,output s[4])**

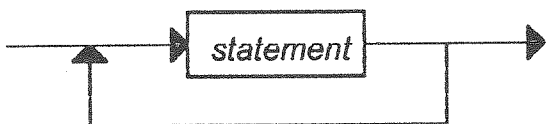
```
{
--Adder4Bit BODY
}
```

شکل (۶) تعریف یک جمع کننده ۴ بیتی در AHDSL

۳-۲. قواعد جملات در بدنه معماری

بدنه هر معماری شامل مجموعه‌ای از جملات است، که با یک نقطه - واوک (semicolon); از یکدیگر جدا می‌شوند.

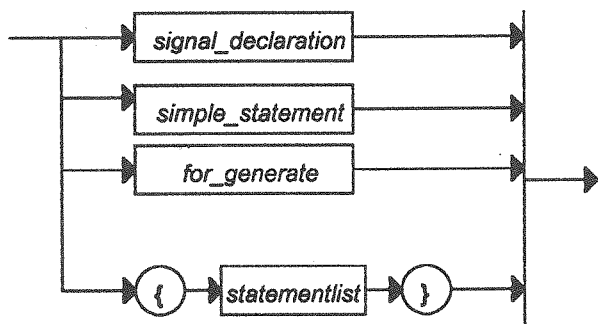
statementlist



شکل (۷) ترکیب جملات در بدنه معماری در برنامه‌های AHDSL

جملات این زبان شامل تعریف متغیرها، جملات نسبت‌دهی و جملات تکرار می‌باشد.

statement



شکل (۸) قواعد انواع جملات مجاز در داخل معماری

و قسمت دوم شامل بدنه است، که از مجموعه‌ای از جملات تشکیل می‌گردد. هر معماری شامل جملاتی است که همگی بصورت همروند (concurrent) اجرا می‌گردند، به همین دلیل ترتیب نوشتن جملات اهمیتی ندارد. در شکل (۳) قاعده تعریف یک معماری نشان داده شده است.

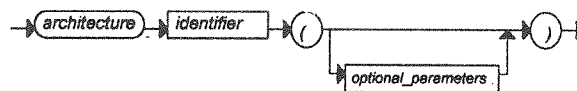
architecture definition



شکل (۳) قاعده تعریف یک معماری در AHDSL

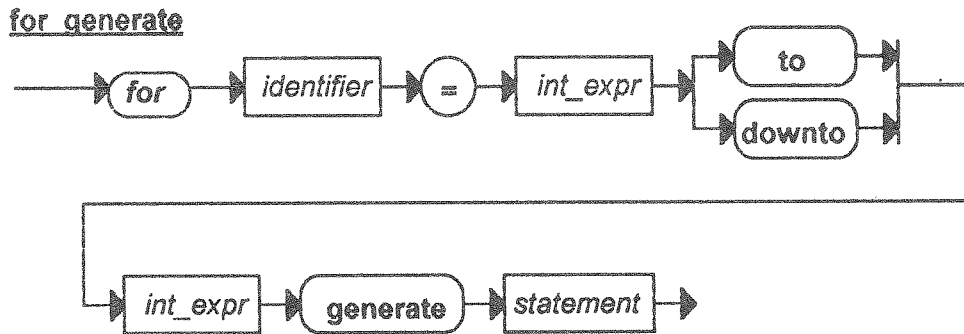
هر معماری با یک لغت کلیدی architecture شروع می‌شود و سپس یک متغیر که نشان‌دهنده نام این معماری است و پارامترهایی که ورودیها و خروجیها را تعیین می‌کنند، معرفی می‌شوند. یک معماری می‌تواند هیچگونه پارامتری نداشته باشد و یا به هر تعداد پارامتر داشته باشد. این پارامترها رابط بین معماری و دنیای خارج هستند. شکل (۴) قاعده معرفی یک معماری را نشان می‌دهد.

architecture head



شکل (۴) قاعده معرفی یک معماری در AHDSL

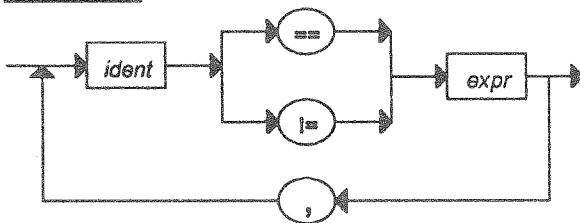
پارامترهای یک معماری می‌توانند یکی از سه حالت ورودی (input)، خروجی (output) و یا ورودی/خروجی (inout) را داشته باشند. در حالتی که یک پارامتر، ورودی باشد پارامتر موردنظر فقط قابل خواندن است و پارامتری که ورودی/خروجی باشد هم قابل نوشتن و هم قابل خواندن است. تفاوت بین پارامترهای خروجی و پارامترهای ورودی/خروجی در این است که روی یک پارامتر خروجی مستقیماً می‌توان نوشت اما یک سیگنال ورودی/خروجی تنها از طریق یک میانگیر سه حالت (tri state) قابل نوشتن است. شکل (۵) قاعده تعریف پارامترها را نشان می‌دهد. برنامه شکل (۶) تعریف یک جمع کننده ۴ بیتی است که دارای ۲ ورودی و ۲ خروجی است.



شکل (۹) قاعده تعریف جملات تکرار در AHDSL

می دانیم که یک گزاره منطقی بطور کلی از شکلی بصورت: "اگر شرط الف و شرط ب و آنگاه" استفاده می نماید. به همین دلیل از جملات فصل (comma - expr) برای نشان دادن شرطها در یک گزاره استفاده می کنیم. هر جمله فصلی، (AND) از مجموعه ای از عبارتهای شرط تشکیل شده است که با یک واوک از یکدیگر جدا شده اند. عبارتهای شرط می توانند مساوی بودن و یا نامساوی بودن را مورد آزمایش قرار دهند.

comma expr



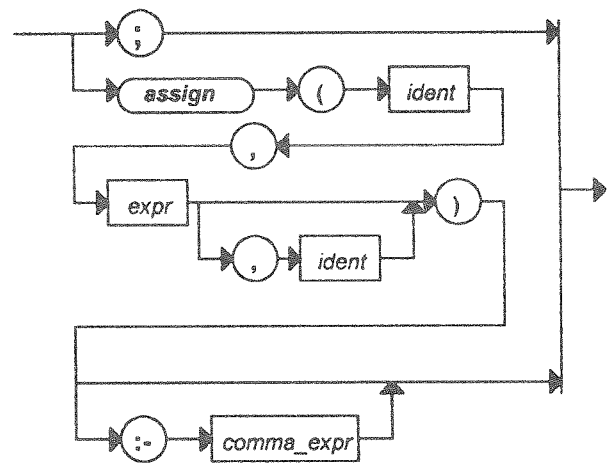
شکل (۱۲) قواعد انواع عبارتهای شرطی در AHDSL

در این زبان دو نوع عبارت وجود دارد. عبارتهای بولین که یک مقدار بولین برمی گردانند و عبارتهای عدد صحیح که یک مقدار عدد صحیح را برمی گردانند. عبارتهای بولین دارای قواعد نشان داده شده در شکل (۱۳) می باشند.

در شکل (۱۴) تعدادی از عبارتهای بولین مجاز در این زبان نشان داده شده است. تابع bin مقدار باینری آرگومان اول خود را به طولی که توسط آرگومان دوم خواسته شده است برمی گرداند. در بسیاری از مواقع لازم است تا از عبارتهای عدد صحیح استفاده نماییم. در AHDSL این عبارتها، همانند قواعد نشان داده

جملات نسبت دهی می توانند بصورت شرطی یا غیر شرطی باشند. همه جملات نسبت دهی از لغت کلیدی assign استفاده می کنند.

simple statement



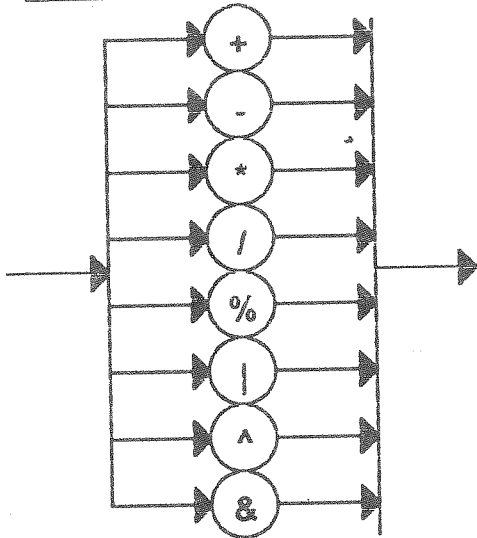
شکل (۱۰) قواعد جملات ساده در AHDSL

جملات زیر انواع مختلف جملات نسبت دهی را نشان می دهند.

```
assign(ASignal,"1");
assign(OtherSignal,ASignal):- ConditionSignal=="1";
assign(OtherSignal,"1"):- ConditionSignal !="1",ResetSignal=="0";
```

شکل (۱۱) نمونه هایی از جملات نسبت دهی در AHDSL

int_operator



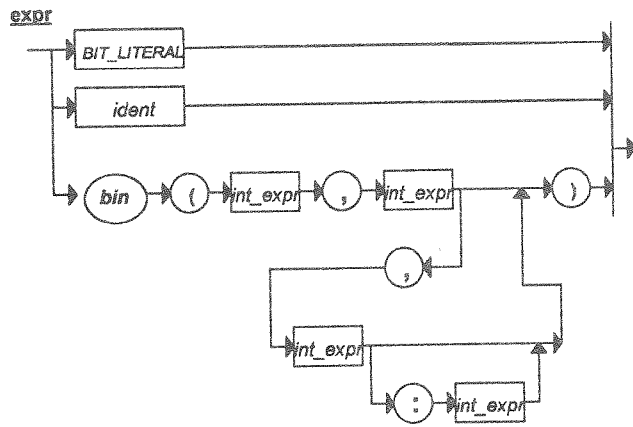
شکل (۱۶) انواع اپراتورهای عدد صحیح مجاز در AHDSL

جملات نسبت‌دهی از روی نوع متغیری که بکار برده می‌شود لاجیک مربوطه را می‌سازند. بطور مثال در شکل (۱۷)، اگر Signal A یک متغیر معمولی باشد آنگاه این جمله به

assign (SignalA,SignalB):-SignalC

=="10" , SignalD[2:3]=="11";

شکل (۱۷) یک جمله نسبت‌دهی، برای توضیح تشخیص لاجیک مورد نیاز از روی نوع متغیرها



شکل (۱۳) قواعد استفاده از عبارتهای بولین در AHDSL

"10011" -- یک مقدار عددی --

SignalA -- یک متغیر بولین --

SignalB[2][3:1] -- یک محدوده از یک متغیر بولین --

bin(20,6) -- معادل ۶ بیتی عدد ۲۰ --

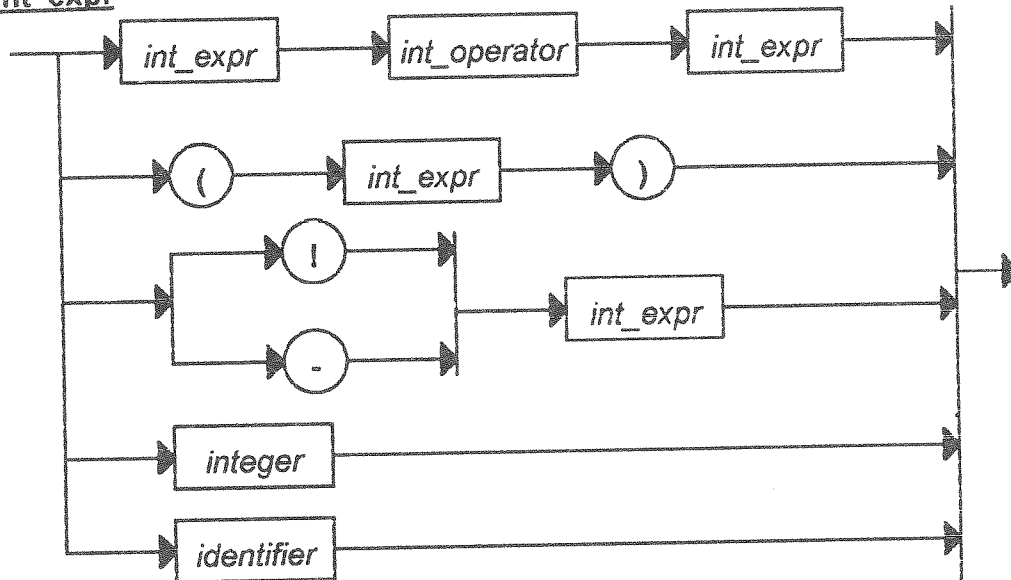
bin(20,6,4) -- ۴ از معادل ۶ بیتی عدد ۲۰ --

bin(20,6,4:2) -- ۲ تا ۴ از معادل ۶ بیتی عدد ۲۰ --

شکل (۱۴) نمونه‌هایی از عبارتهای بولین

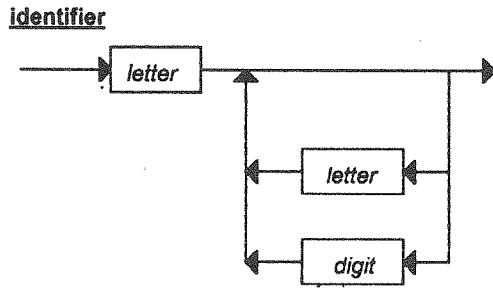
شده در پایین از عملگرهای OR, AND, XOR, NOT باقیمانده "%", "خارج قسمت"/, "-", و "+" استفاده می‌نمایند.

int_expr



شکل (۱۵) قواعد عبارتهای عدد صحیح در AHDSL

نشان داده شده است.



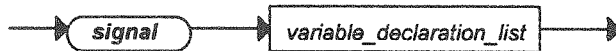
شکل (۲۱) قاعده تعریف شناسه در AHDSL

در شکل (۲۱) حروف (letter) شامل A-Z ، a-z و '-' است و رقم (digit) شامل 0-9 است. اعداد صحیح در این زبان می‌توانند مقادیر از $2^{31}-1$ تا -2^{31} را در برگیرند. قواعد اعداد صحیح بصورت مبنای ۱۰ و یا مبنای ۱۶ نوشت.

۲-۴- قواعد تعریف متغیرها

تعریف یک سیگنال با لغت کلیدی Signal شروع شده و شامل تعریف متغیر با انواع مختلف است. این متغیرها با واوک از یکدیگر جدا می‌گردند و در انتها نیز تعریف سیگنال با یک نقطه واوک (Semicolon) تمام می‌گردد. تعریف متغیرها می‌تواند در هر خطی از برنامه قرار داشته باشد، ولی هر متغیر قبل از استفاده باید تعریف شده باشد.

signal declaration



شکل (۲۲) قاعده تعریف متغیرها

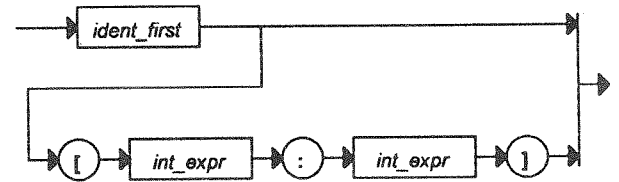
انواع متغیرهایی که می‌توان در این زبان تعریف نمود عبارتند از:
 الف) سیگنالهای اجزاء حافظه‌دار (memo): این سیگنالها اجزایی مانند فلیپ فلاپها و رجیسترها را تعریف می‌نمایند.
 ب) سیگنالهای اجزاء گذرگاهها (tri): این سیگنالها نشان‌دهنده باسها در یک معماری هستند.
 ج) سیگنالهای ساختگی (dummy): سیگنالهای ساختگی در زمان ترجمه توسط مترجم حذف خواهند شد. به همین دلیل

امیرکبیر / سال هشتم / شماره ۲۹

تعدادی And, OR تبدیل می‌گردد. اما اگر Signal A بعنوان یک باس (INOUT یا TRI) باشد، آنگاه یک بافر سه حالت ایجاد خواهد شد که قسمت شرط، کنترل آن را تشکیل داده و Signal B نیز ورودی آن خواهد بود. اما اگر Signal A یک عنصر حافظه‌دار (MEMO) باشد، آنگاه فلیپ فلاپ مناسب ایجاد خواهد شد.

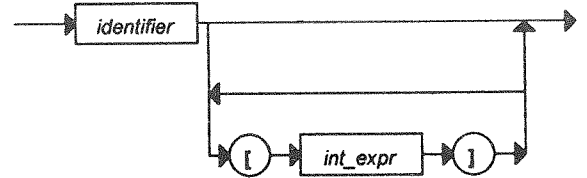
در این زبان می‌توان به یک متغیر به روشهای گوناگون دستیابی داشت. قواعد نحوه دستیابی به متغیرها به صورت نشان داده شده در شکل (۱۸) و شکل (۱۹) می‌باشد. نمونه‌ای از نحوه رجوع به متغیرهای این زبان در شکل (۲۰) آورده شده است.

ident



شکل (۱۸) قواعد دستیابی به متغیرها در AHDSL

ident first



شکل (۱۹) ادامه قواعد دستیابی به متغیرها در AHDSL

```
assign(SignalA,bin(0,4*3))-SignalB[3]=="1";
assign(SignalA[5][7:6],"11"):-SignalB[2:0]=="111";
assign(SignalA[5][0],SignalC[7]):-SignalC==bin(3,8);
```

شکل (۲۰) نحوه رجوع به متغیرها

همانطور که دیده می‌شود در جمله اول به تمامی SignalA و در جمله بعدی به یک قسمت و در جمله آخر تنها به یک بیت مقدار دهی شده است.

شناسه در این زبان با یک حرف شروع شده و سپس با حرف یا رقم ادامه می‌یابد. هیچ محدودیتی برای طول شناسه‌ها وجود ندارد. در شکل (۲۱) قواعد نوشتن شناسنامه در AHDSL

نمونه‌هایی از متغیرها که در این زبان می‌توان تعریف نمود در شکل (۲۶) نشان داده شده است.

signal

**memo AFlipFlop, memo ARegister[8],
-- memory signals**

tri ABus[16], -- a bus signal

dummy ATemp[10], -- temporary signal

ASignal; یک متغیر معمولی

شکل (۲۶) مثالی از تعریف متغیرهای مختلف در AHDSL

۳- توصیف پردازنده PARWAN با AHDSL

پردازنده PARWAN یک پردازنده ۸ بیتی ساده است که برای مقاصد آموزشی طراحی شده است [۳]. این پردازنده دارای یک باس داده ۸ بیتی و باس آدرس ۱۲ بیتی است و به این ترتیب می‌تواند ۴۰۹۶ بایت حافظه را آدرس‌دهی نماید. مدلی که در اینجا مورد بررسی قرار می‌گیرد دارای ۲۳ دستورالعمل بوده و از دو مد آدرس‌دهی حافظه استفاده می‌نماید.

۳-۱- ساختار داخلی PARWAN

ساختار داخلی PARWAN در شکل (۲۷) نشان داده شده است. در این شکل روشی که رجیسترها و اجزاء لاجیک در کنار هم قرار گرفته‌اند نشان داده شده است. این شکل برای توصیف جزئیات داخلی PARWAN برای یک توصیف جریان داده‌ها (DATA FLOW) مفید است. اجزاء اصلی PARWAN عبارتند از:

AC: یک رجیستر ۸ بیتی همه منظوره است که نتیجه اعمال منطقی و حسابی در این رجیستر نوشته می‌شود. این پردازنده تنها یک رجیستر همه منظوره دارد.

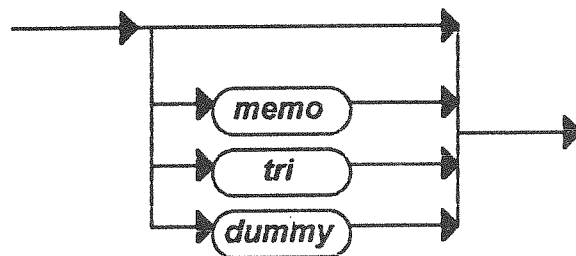
IR: یک رجیستر ۸ بیتی است که در مرحله واکنشی (fetch) با کد دستورالعملی که باید اجرا گردد پر می‌شود.

PC: این پردازنده دارای یک شمارنده برنامه ۱۲ بیتی است که ۴ بیت بالای آن شماره صفحه و ۸ بیت پایین، مقدار جبرانی در داخل صفحه را تعیین می‌نماید. پردازنده PARWAN دارای دو مد آدرس‌دهی حافظه است: مد آدرس کامل (Full Address) که در آن تمامی آدرس توسط دستورالعمل تعیین می‌شود، و مد

می‌توان از آنها برای ساده‌تر شدن توصیف استفاده نمود. بطور مثال می‌توان یک جمع‌کننده رقم نقلی را تنها توسط تعریف رقمهای نقلی بعنوان سیگنال ساختگی به یک جمع‌کننده موازی تبدیل نمود که تمامی رقمهای نقلی با تأخیر یکسان آماده گردند. (د سیگنالهای معمولی: این سیگنالها تنها بصورت مجازی وجود داشته و بطور کلی نشان‌دهنده مدارات ترکیبی هستند که بین دو گره از مدار وجود دارند).

هر سیگنال می‌تواند یکی از خصوصیات، memo برای نشان دادن فلیپ فلاپها، tri برای نشان دادن گذرگاهها، dummy برای نشان دادن سیگنالهای ساختگی و یا معمولی، را دارا باشد.

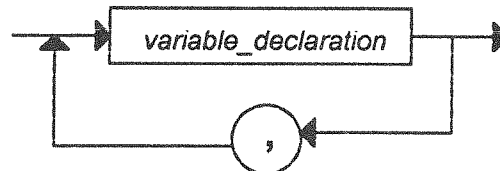
declarator



شکل (۲۳) قاعده تعریف انواع متغیرهای مختلف

یک مجموعه از متغیرها که بوسیله واوک از یکدیگر جدا شده‌اند یک لیست از متغیرها را تشکیل می‌دهند.

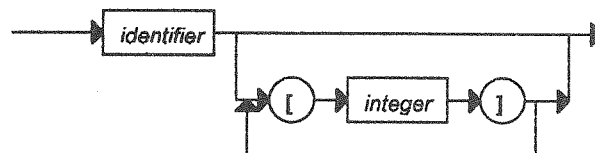
variable declaration list



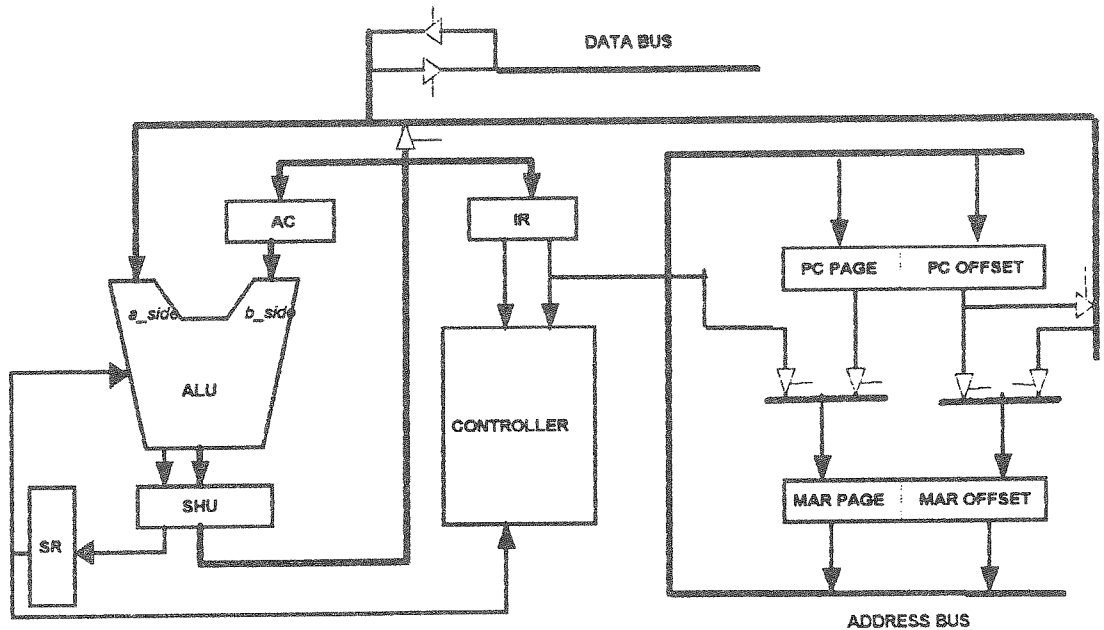
شکل (۲۴) قاعده تعریف یک لیست از متغیرها

هر متغیر بصورت یک آرایه از بیتها تعریف می‌شود که این آرایه می‌تواند دارای چندین بعد باشد.

variable declaration



شکل (۲۵) قاعده تعریف یک متغیر



شکل (۲۷) ساختار داخلی PARWAN

S2	S1	S0	OPERATION
0	0	0	a AND b
0	0	1	NOT b
1	0	0	a
1	0	1	b PLUS a
1	1	0	b
1	1	1	b MINUS a

شکل (۲۸) اعمال ALU

حالت، سیگنالهای کنترل مناسب را برای قسمت‌های مختلف ایجاد می‌نماید.

در بخش‌های باقیمانده به توصیف پردازنده بوسیله AHDSL خواهیم پرداخت.

۲-۳- توصیف اینترفیس پردازنده

سیگنالهایی که مورد استفاده قرار می‌گیرند عبارتند از:

clk: یک ورودی بوده و پالس ساعت سیستم است. تمامی اعمال انجام شده بوسیله این پالس با یکدیگر سنکرون می‌گردند.

آدرس‌دهی در صفحه، که در این مد تنها مقدار جبرانی توسط دستورالعمل مشخص شده و صفحه توسط ۴ بیت بالای شمارنده برنامه تعیین می‌گردد.

SR: این رجیستر یک رجیستر ۴ بیتی است که به ترتیب پرچم‌های سرریز، رقم نقلی، صفر و منفی را در خود نگه می‌دارد. پرچمها به طور غیرمستقیم و تحت تأثیر مقداری که در AC نوشته می‌شود تغییر می‌کنند. البته پرچم رقم نقلی توسط دستورالعمل CMC نیز قابل تغییر است.

MAR: این رجیستر آدرس محلی از حافظه را که توسط پردازنده مورد دستیابی قرار می‌گیرد، در خود نگه می‌دارد. همانطور که قبلاً نیز گفته شد، PARWAN از روش صفحه‌بندی به حافظه دست می‌یابد و به همین دلیل MAR نیز از دو قسمت تشکیل شده است:

ALU: این قسمت یک مدار ترکیبی است که می‌تواند اعمال لیست شده در شکل (۲۸) را انجام دهد. S سیگنال کنترل ALU است.

SHU: این قسمت برای تکمیل وظیفه ALU استفاده می‌شود زیرا ALU قادر به انجام عمل جابجایی (shift) روی بیتها نمی‌باشد.

CONTROL: کنترل این پردازنده با روش فلیپ فلاپ برای هر حالت (one flip flop per state) دارای ۹ حالت است که هر

می‌شود که محتویات رجیستر PAGE—PC روی باسی که به ورودی PAGE—MAR وصل است قرار گیرد.

```
signal tri DBus[8], tri MarPageBus[4],
      tri MarOffsetBus[8];
signal IROnMarPageBus, PcPageOnMarPageBus,
      PcOffsetOnMarOffsetBus;
signal PcOffsetOnDBus, DBusOnMarOffsetBus,
      ShuOutOnDBus, DataBusOnDBus,
      DBusOnDataBus;
assign(DBus,data):-DataBusOnDBus=="1";
assign(data,DBus):-DBusOnDataBus=="1";
assign(MarOffsetBus,DBus):-DBusOnMarOffsetBus
      ==="1";
```

شکل (۳۰) توصیف باسهای داخلی و سیگنالهای کنترل مربوطه با AHDSL

۴-۳- توصیف قسمت کنترل پردازنده

در توصیف کنترل پردازنده از روش یک فلیپ فلاپ برای هر حالت استفاده شده و جمعاً ۹ حالت مجزا داریم. این قسمت نمونه خوبی از یک ماشین حالت است که یکی از مهمترین مسایل در طراحی مدارهای منطقی است. در این مقاله تنها توصیف حالت آغاز مجدد و حالت اول پردازنده نشان داده شده است.

کنترلر در هر حالتی که باشد در صورتی که سیگنال intr برابر یک شود به حالت اول برمی‌گردد. در شکل (۳۱) نحوه انجام این عمل نشان شده است.

```
signal memo CtrState[9];
- RESET STATES TO ALL ZERO.
assign(CtrState,"00000000",clk);
assign(CtrState,"00000001",clk):-intr=="1";
```

شکل (۳۱) برنامه آغاز مجدد پردازنده با AHDSL

۵-۳- توصیف حالت اول

در این حالت سیستم در صورتی که intr برابر یک باشد سیگنال PcReset را یک می‌نماید که این امر باعث می‌گردد که با پالس بعدی مقدار PC رجیستر برابر صفر شود و همینطور بدون در نظر گرفتن سیگنال intr محتویات شمارنده برنامه در MAR رجیستر بار می‌گردد.

intr: یک ورودی یک بیتی است که اگر برابر یک شود باعث reset شدن سیستم می‌شود. این پردازنده پس از reset شدن از آدرس 000HEX به کار خود ادامه می‌دهد.

data: که بصورت یک باس تعریف شده است و ۸ بیتی است. read: یک سیگنال یک بیتی است و در هنگامی که پردازنده عمل خواندن از حافظه را انجام می‌دهد این سیگنال یک می‌شود.

write: یک سیگنال یک بیتی است و در هنگامی که پردازنده عمل نوشتن به حافظه را انجام می‌دهد این سیگنال یک می‌شود. addr: خروجی بوده و ۱۲ بیتی است. آدرسی از حافظه را که باید عمل خواندن یا نوشتن در آن صورت گیرد روی این سیگنال قرار می‌گیرد.

در شکل (۲۹) توصیف اینترفیس PARWAN بوسیله AHDSL نشان داده شده است.

```
architecture ParWan(input clk,input intr,inout data[8],
      output read, output write,output addr[12])
```

-- PARWAN BODY

شکل (۲۹) توصیف اینترفیس PARWAN با AHDSL

۳-۳- توصیف باسهای داخلی و سیگنالهای کنترل مربوطه

این پردازنده دارای ۳ باس داخلی است که عبارتند از:

DBus: یک باس ۸ بیتی است که به ورودی a—side، ALU، MAR—OFFSET و DATA—BUS وصل است و خروجیهای SHU—OFFSET و PC—OFFSET را می‌توان روی این باس قرار داد.

MarPageBus: یک باس ۴ بیتی است و به ورودی PAGE—MAR وصل است و خروجیهای PC—PAGE و ۴ بیت پایین IR را می‌توان روی آن قرار داد.

MarOffsetBus: یک باس ۸ بیتی است و به ورودی MAR—OFFSET وصل است و خروجیهای PC—OFFSET و DBUS را می‌توان روی آن قرار داد.

سیگنالهای مورد استفاده وقتی که یک می‌شوند باعث می‌گردند که مقدار سیگنال سمت چپ روی باس سمت راست قرار گیرد. بطور مثال سیگنال PcPageOnMarPageBus باعث

کار مشابهی به کمک زبان VHDL برای توصیف PARWAN انجام شده است. مدار ساخته شده از توصیف VHDL جمعاً از ۱۲۳۱ گیت استفاده می‌نماید که اگر بطور متوسط ۵ ترانزیستور برای هر گیت در نظر بگیریم جمعاً ۶۱۵۵ ترانزیستور در توصیف VHDL این پردازنده مصرف شده است. زمان ترجمه برای این توصیف با یک کامپیوتر ۳۸۶، ۳۳ مگاهرتز با ۴ مگابایت حافظه در حدود ۶ ساعت بوده است. مترجم AHDSL قادر است توصیف PARWAN را با یک کامپیوتر ۳۸۶، ۳۳ مگاهرتز با ۴ مگابایت حافظه در کمتر از ۳۰ ثانیه ترجمه نموده و دو فایل ایجاد نماید. فایل اول شامل فهرست دریاچه‌ها است و فایل دوم نیز یک فایل VHDL است که می‌توان از آن برای شبیه‌سازی نتایج استفاده نمود. همینطور از این فایل می‌توان برای تخمین پیچیدگی مدار نیز استفاده کرد. با بررسی این فایل برای توصیف فوق طرح نهایی حدوداً دارای ۴۰۰۰ ترانزیستور می‌باشد. شکل (۳۳) مقایسه نتیجه حاصل از توصیف PARWAN با VHDL و AHDSL را نشان می‌دهد.

	VHDL	AHDSL
Compile Time	6 Hour	30 Sec
No of Transistors	6155	4000

شکل (۳۳) مقایسه نتیجه حاصل از توصیف PARWAN با AHDSL و VHDL

-STATE #1 DESCRIPTION.

```
assign(PcPageOnMarPageBus,"1"):-
  CtrState[0]=="1";
assign(PcOffsetOnMarOffsetBus,"1"):-
  CtrState[0]=="1";
assign(MarLoadPage,"1"):-CtrState[0]=="1";
assign(MarLoadOffset,"1"):-CtrState[0]=="1";
assign(CtrState[1],"1",clk):-
  CtrState[0]=="1",intr=="0";
assign(PcReset,intr):-CtrState[0]=="1";
```

شکل (۳۲) توصیف حالت اول با AHDSL

۴- نتیجه‌گیری

با توضیحات ارائه شده می‌توان دید که زبان توصیف سخت افزار AHDSL دارای قابلیت‌های مشروح ذیل است:

(الف) بوسیله AHDSL می‌توان مدارات دیجیتال مختلف از حافظه‌ها، شمارنده‌ها، ماشین‌های حالت، پردازنده‌ها و ... را با ساختارهای ساده توصیف نمود.

(ب) امکان توصیف یک مدار تکراری (iterative network) بصورت سریال و استفاده از سیگنال‌های ساختگی برای تبدیل توصیف به صورت موازی را دارا می‌باشد.

(ج) AHDSL دارای گرامر بسیار ساده‌ای است و براحتی قابل یادگیری است و به همین دلیل در زمینه آموزش مفاهیم سخت افزار بسیار کارآمد می‌باشد.

مراجع

- [1] Nikii. D. Dutt and Daniel. D. Gajski, "Design Synthesis and Silicon Compilation", IEEE Trans. Design & Test OF Computers, Dec. 1990 PP. 8-22.
- [2] Giovannie De Micheli, "High Level Synthesis of Digital Circuits" IEEE Trans. Design & Test OF Computers, Oct. 1990. PP 6-7.
- [3] Navabi, Z., "VHDL : Analysis and Modeling of Digital Systems," Mc Graw-Hill Publishing NewYork, 1991.
- [4] خسروی، مسعود، "پیاده‌سازی مترجم یک زبان توصیف سخت‌افزار"، دانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر، شهریور ۱۳۷۴.