Hardware Interchange Format

https://github.com/masc-ucsc/hif

Ramesh Jayaraman, Jing-Hsiang Huang, and Jose Renau
Department of Computer Science and Engineering
University of California, Santa Cruz

Motivation

- Many open-source hardware tools and compilers exist
- It would be useful to interchange between all of them
- There are commonalities among them
  - IRs use Trees/Netlists

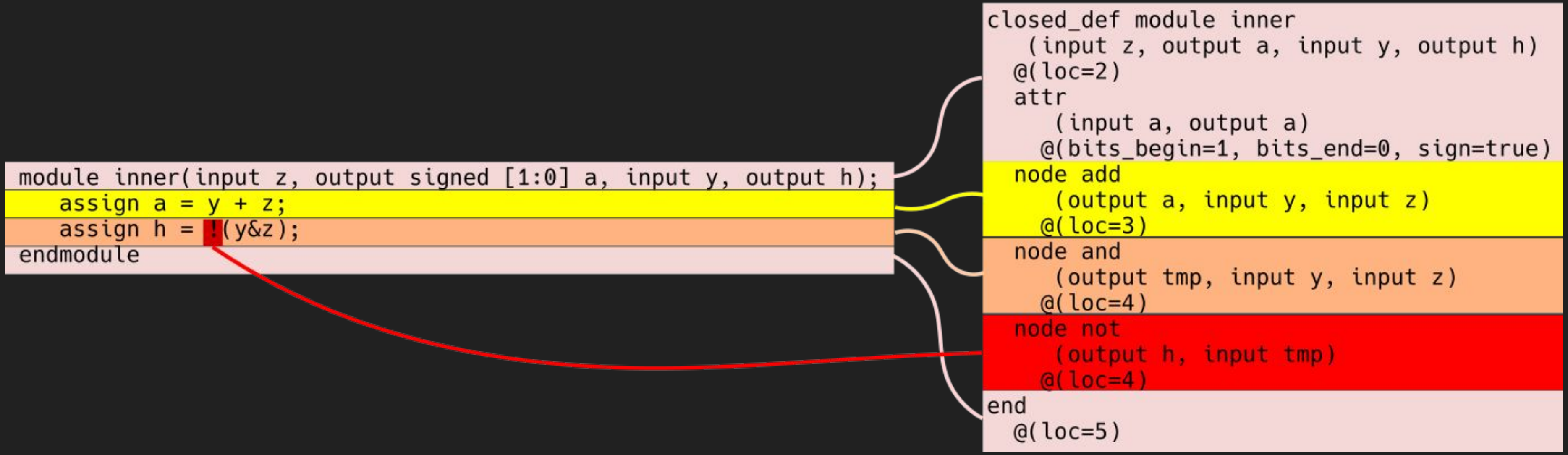
The Hardware Interchange Format

- Common API
- Serialize/Deserialize hardware IRs
- Fast
- Efficient
- Standardized
    - Implemented to be a common Library
    - Uses C++ - 17 standard

Data Format

- Comprises of statements of different types
    - Stores graph and tree properties common in IRs
- Each statement
    - Has a class (extendable to custom classes per compiler)
    - Generic Fields (Type, I/O, etc.)
    - Optional Fields (Specific to languages)

HIF Example



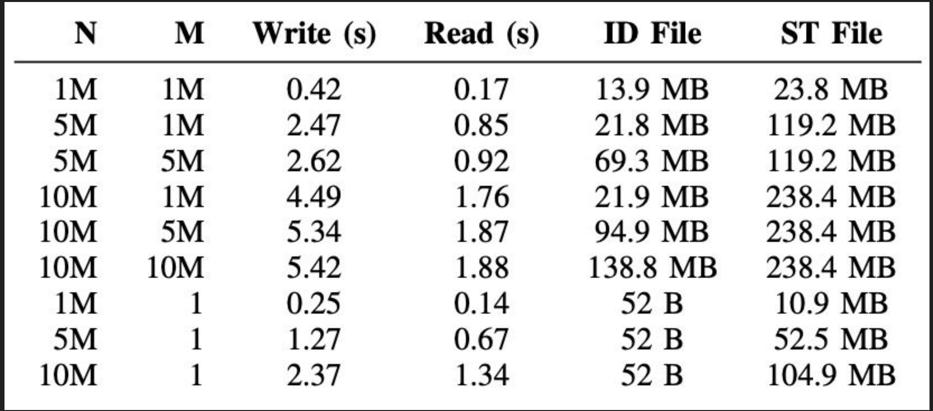
```
module inner(input z, output signed [1:0] a, input y, output h);
   assign a = y + z;
   assign h = !(y&z);
endmodule
```

```
closed_def module inner
   (input z, output a, input y, output h)
  @(loc=2)
  attr
     (input a, output a)
    @(bits_begin=1, bits_end=0, sign=true)
  node add
     (output a, input y, input z)
    @(loc=3)
  node and
     (output tmp, input y, input z)
    @(loc=4)
  node not
     (output h, input tmp)
    @(loc=4)
end
  @(loc=5)
```

Binary Encoding

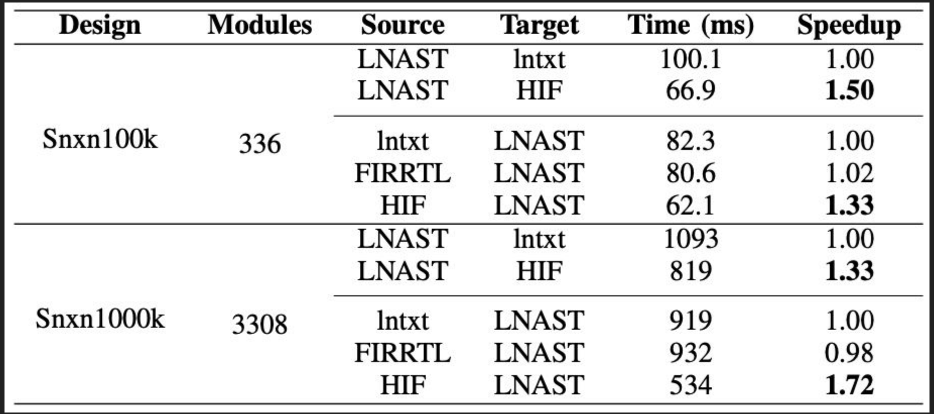
- Efficient encoding in binary files
- Stores data in two sets of files for efficient access
  - Identifiers:
    - Alphanumeric constant with support for multiple languages
  - Statements:
    - Structured encoding using statement type

Results



N	M	Write (s)	Read (s)	ID File	ST File
1M	1M	0.42	0.17	13.9 MB	23.8 MB
5M	1M	2.47	0.85	21.8 MB	119.2 MB
5M	5M	2.62	0.92	69.3 MB	119.2 MB
10M	1M	4.49	1.76	21.9 MB	238.4 MB
10M	5M	5.34	1.87	94.9 MB	238.4 MB
10M	10M	5.42	1.88	138.8 MB	238.4 MB
1M	1	0.25	0.14	52 B	10.9 MB
5M	1	1.27	0.67	52 B	52.5 MB
10M	1	2.37	1.34	52 B	104.9 MB

Results



Design	Modules	Source	Target	Time (ms)	Speedup
Snxn100k	336	LNAST	lntxt	100.1	1.00
		LNAST	HIF	66.9	**1.50**
		lntxt	LNAST	82.3	1.00
		FIRRTL	LNAST	80.6	1.02
		HIF	LNAST	62.1	**1.33**
Snxn1000k	3308	LNAST	lntxt	1093	1.00
		LNAST	HIF	819	**1.33**
		lntxt	LNAST	919	1.00
		FIRRTL	LNAST	932	0.98
		HIF	LNAST	534	**1.72**

Conclusions

Available at https://github.com/masc-ucsc/hif

Used by LiveHD High level and low level IRs

Several tasks/tools can be build/leverage it

Thanks,