MyHDL: Fixed-point Compiler

Sub-org

MyHDL

Personal Info

Name	***
GitHub username	qrqiuren
Email	***
Tel	***
Timezone	Tokyo (UTC+0900)
Blog RSS	http://qren.cc/posts.atom (newly set up for English blogging, will write for GSoC 2017)

Code Sample

Here are some open source code samples in my past work that are somehow related to this project.

MyHDL

I created a pull request to fix an issue in the back-end. This PR is required by MyHDL project in order to take part in GSoC.

particle

This is my undergraduate thesis project. I used NumPy to perform signal processing in this project and made detailed comments in the code. My graduation thesis (in Chinese) is the documentation of this repository.

Project Info

Proposal title

MyHDL: Fixed-point Compiler

Proposal Abstract

This project is aimed to provide a fixed point compiler backend for the `fixbv` type defined in [MEP-111](#) of MyHDL project.

When the project finishes, MyHDL should have a complete implementation of `fixbv` and its compiler backend with complete documentation, include tutorials and reference.

Proposal details

Since fixed point numbers are heavily used in applications such as signal processing, it is important to implement it in MyHDL to make it convenient for users.

In [MEP-111](#), a fixed point type `fixbv` has been defined in order to provide fixed point type implementation in MyHDL. The highlight of MEP-111 is that the `fixbv` type inherits the `intbv` which is much mature in MyHDL. It will simplify the implementation of `intbv`.

At the end of this project, the fixed point type `fixbv` should be implemented as described in MEP-111 along with detailed documentation, including reference and tutorials.

These are what I am considering during making the schedule:

- Make a backup week every month in order to (1) leave time for code cleanup (2) prevent potential changes of my schedule, especially those of my lab projects. (My lab is designing ICs and we will be busy before the deadline of tape out)
- I have marked the known important things in my own schedule that may lag the progress of GSoC.
- Make a documentation week in each month. It is better to document the code soon after it finishes.
- Write a post in [my blog](#) each week, which is not shown in the schedule. This is very important to make the progress visible and share my experience to people.
- The due date of each week is set on Fridays.

Here is a schedule of the planning project.

Wk#	Due	Plan
1	May 12	Greetings to people (Lab T.O. deadline on May 10)
2	May 19	• Read existing code in the repo • Play around the code and related libraries

3	May 26	- Same as Wk#2 - Review the schedule with mentor
4	Jun 2	- Check with mentor if existing work on MEP-111 is available for my project - Complete an interface of `fixbv` class
5	Jun 9	Implement `fixbv` class and write test cases (may leave a few difficult methods not implemented)
6	Jun 16	Documentation for `fixbv` class
7	Jun 23	- Check up the existing code in compiler backend - Make a to-do list for changes that should be make in compiler backend, especially those are unique to `fixbv`
8	Jun 30	Backup (Important exam on July 2)
9	Jul 7	Write test cases of arithmetics for `fixbv`
10	Jul 14	Implement arithmetics for `fixbv`
11	Jul 21	Documentation
12	Jul 28	Backup
13	Aug 4	Kill the remaining tasks in the to-do list
14	Aug 11	Write test cases for the implemented parts
15	Aug 18	- Write sample tutorials - Final documentation check
16	Aug 25	Final clean-up
17	Aug 30	Backup

Other Commitments

My lab will have a tape out deadline on May 10, I will be extremely busy during that period (even no time to go home!).

I will take an important exam of Japanese language level on July 2. This is important for my scholarship application in the future.

I am going to take a business training program in the university from the end of April to July. However, I am considering to quit because it may have conflicts to my other schedule in my university, and I want to save time for GSoC.

If there are some other things really important that may affect the schedule, I will contact my mentor to adjust the schedule in that case.

I have applied for AerospaceResearch.net and JSK. I would prefer JSK first, then Aerospace, then PSF.

Extra Info

Resume	[Click here](#)
University	Tokyo Institute of Technology
Major	Graduate Major of Electric and Electronic Engineering
Current year	1st year Master student, expected to graduate in Sep 2018
Degree	Master of Engineering (?)
Twitter	@qrqiuren
IM	***
Related PR	[myhdl/myhdl#217](#)