# Review

- Set up the environment successfully.

- Use nvprof in the command line, and write a script to run nvprof of a program. But this is to run the whole program in real GPU hardware, not close to our purpose.

- CUDA programming example, I do not know how to run some part in GPGPU-Sim and some in GPU automatically. Just write in the program , some is sequential , some is parallel, and run the whole program.

- CUPTI, this is not so familiar
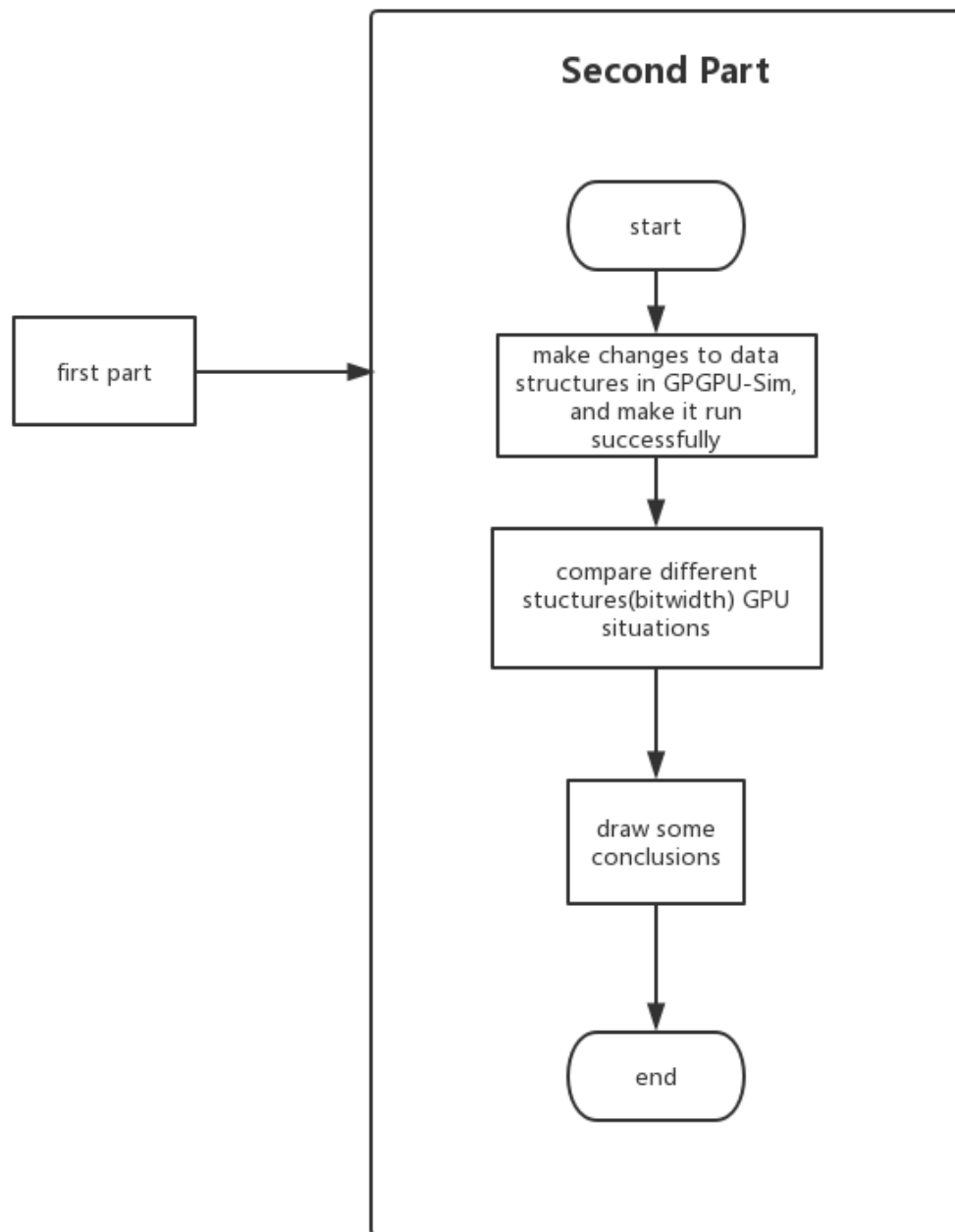
Questions(new version)

Now my understanding of our first part：
1. Use CUPTI to trace the situation of GPU
2. Collect the information and recognize them
3. Transfer the information to GPUWattch
4. Use GPUWattch to ouput the information
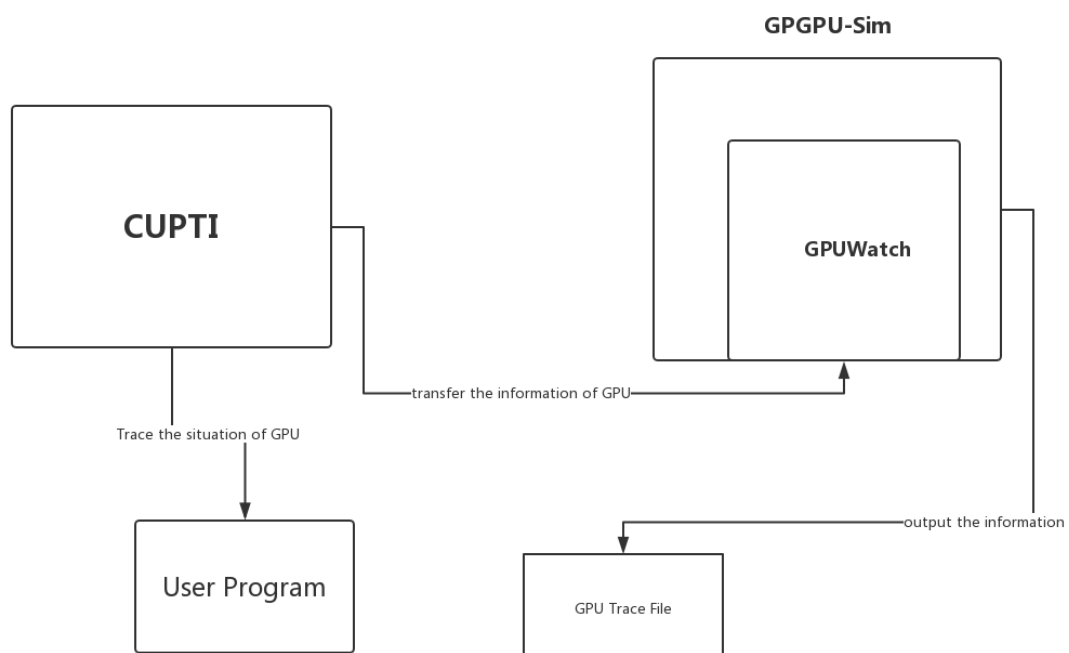
The overall understanding of our project:

Second Part

```
                    ┌─────────┐
                    │  start  │
                    └─────────┘
                         │
                         ▼
┌───────────┐    ┌──────────────────────┐
│ first part│───▶│ make changes to data │
└───────────┘    │ structures in        │
                 │ GPGPU-Sim,           │
                 │ and make it run      │
                 │ successfully         │
                 └──────────────────────┘
                         │
                         ▼
                 ┌──────────────────────┐
                 │ compare different    │
                 │ stuctures(bitwidth)  │
                 │ GPU situations       │
                 └──────────────────────┘
                         │
                         ▼
                 ┌──────────────────────┐
                 │ draw some            │
                 │ conclusions          │
                 └──────────────────────┘
                         │
                         ▼
                    ┌─────────┐
                    │   end   │
                    └─────────┘
```

Difficulty(new version)

If we follow the mind, then we do not need to compile and link the program dynamically during the process. All the program is run in GPU, all that we change is to use GPUWattch as a tool to output information about GPU using. **So, the most important part is to use CUPTI code to output the GPU information and transfer it to GPUWattch in the first part.**

Plan(New Version)

Instead nvprof, use CUPTI to trace the GPU using.
Read the CUPTI manual, explore more about GPU in it.
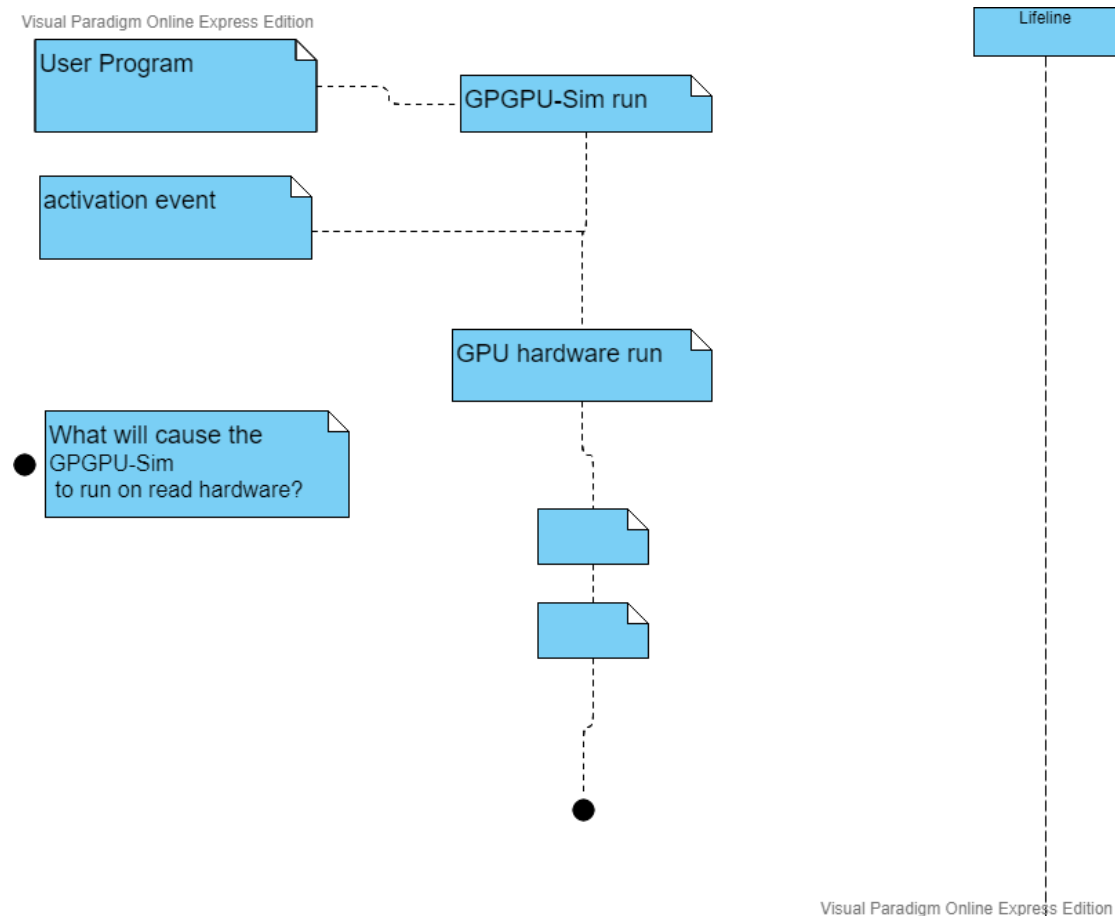Rebuild GPGPU-Sim as an practice, and make changes to it.

Plan(Old Version)

Get more familiar with CUDA programming, so that it will help me understand GPGPU-Sim more and enable me to modify GPGPU-Sim

Find a way to run a program in different commands, some in GPGPU-Sim and some in GPU, this is the most difficult, and I do not know how long it will take

To read some papers about GPGPU-Sim and GPUWatch, to see the interface, whether we can modify it to support different data structure

Questions（old version for reference）

User Program

GPGPU-Sim run

activation event

GPU hardware run

What will cause the
GPGPU-Sim
to run on read hardware?

Lifeline

What will cause the GPGPU-Sim to switch to real GPU hardware?

Run some operations on real GPU, some operations on GPGPU-Sim(approximate multiplier)
Or Run all the operations on real GPU, change the approximate to exact?

Difficulty(Old version)

To run some parts of a program in GPGPU-Sim , some parts in real GPU hardware. Meanwhile, use nvprof or CUPTI to trace the situation of GPU.

Define new data structure(FP) to run the GPGPU-Sim part, compare the result.