FJLIB

FJLIB is an alpha version of the fork-join scheduling library. This folder contains the necessary files to make and compile the library. The library can be configured for certain fork-join tasks by modifying `config.h` and adjusting the number of sections, and max section of threads accordingly. After doing so the library can be compiled with the, `makefile`, within the directory. The only necessasry command needed to run in this directory is:

```
make
```

Folder Contents

`asm` - Assembly files necessary to compile the library. Contains crt0.s a custom c runtime assembly file.

`inc` - Include folder containing necessary headers to compile the library. Most headers contain just prototypes or declarations for their respective file. The `config.h` will need to be adjusted depending on the task and library recompiled.

`src` - all c source files used to compile the library.

- `ep.c` - pointer functions to each core entry point.
- `ep_control.c` - File that handles entry point controls.
- `ep_core1.c - ep_core7.c` - All entry points for all cores. Each entry point will be linked at compile time with their respictive task objects.
- `ep.c` - handles when each core schedule should be dispatched.
- `image.c` - Reads each cores stack pointer to find the entrypoint of each core. The stacks were setup through `crt0.s`
- `lock-step.c` - Handles the locking and releasing of each core.
- `memcpy.c` - Copys memory from a location of n bytes.
- `memset.c` - Fills a memory location with n bytes.

`makefile` - makefile that builds the library.

`README.md` - source of this file.

`riscv32-virt.ld` - custom linker script that places the objects and library in their own space of memory.