To synchronize the multiple treads, a condition variable was used. When the maximum number of threads has been created, the main thread waits for the condition variable to be signaled. Before a thread terminates, it signals the condition variable. When this happens, the main thread is alerted and continues. This way busy waiting is avoided.

A mutex was used to lock the buffer. Before the main or a thread accesses the buffer, it tries to lock the mutex. The main thread acquires the lock before setting up a new thread and releases it just before creating the thread. The threads acquire the buffer when starting and release it before signaling the condition variable.

Inspiration was taken from (Caf, 2011)[1] about the use of condition variables. Deadlock issues were solved with the help of (Cat, 2010)[2].

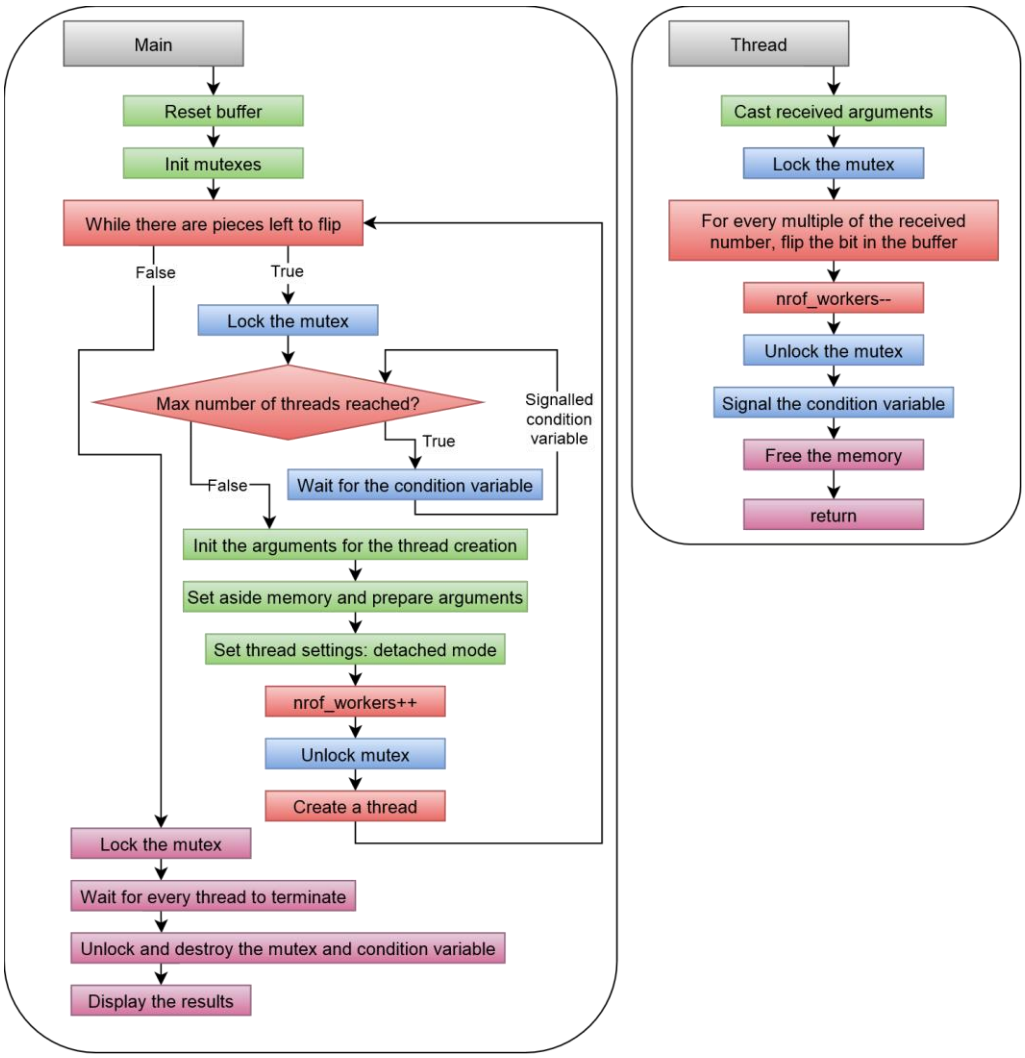

Figure 1 High level flow chart flip.c

[1] Caf. (2011, March 1). Keeping number of threads constant with pthread in C. Retrieved December 9, 2020, from https://stackoverflow.com: https://stackoverflow.com/questions/5150630/keeping-number-of-threads-constant-with-pthread-in-c

[2] Cat. (2010, December 10). Calling pthread_cond_signal without locking mutex. Retrieved December 10, 2020, from https://stackoverflow.com: https://stackoverflow.com/questions/4544234/calling-pthread-cond-signal-without-locking-mutex