2024 Digital IC Design

Homework 3: matrix multiplier

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| NAME | | 楊晴雯 | | | | | | | |
| Student ID | | P76114511 | | | | | | | |
| **Simulation Result** | | | | | | | | | |
| Functional simulation | **100** | | Gate-level simulation | **100** | | Clock  width | 18 (ns) | Gate-level simulation time | 87156  (ns) |
|  | | | | |  | | | | |
| **Synthesis Result** | | | | | | | | | |
| Total logic elements | | | | | 534/55,856 (<1%) | | | | |
| Total memory bit | | | | | 0 (0%) | | | | |
| Embedded multiplier 9-bit element | | | | | 4/308(1%) | | | | |
|  | | | | | | | | | |
| **Description of your design** | | | | | | | | | |
| For this assignment, I designed 2 modules to perform the matrix multiplication.  (1) MMHelper: it handles the writing and computing of data.  (2) MM: it handles when to write data, to which location (which matrix and at which index pair) to write data into, when to compute and all the valid/busy signals.  MM follows the design of “2 Comb + 1 Sequential logic” rules. It uses 5 states to monitor the whole program flow. Each time a testcase (a computation procedure) ends, it goes to RESET state to clear all the signals and the matrix storage inside MMHelper. I move all signals that only change at certain states to the sequential block to ensure the cleanness of the other 2 combinational circuits. Note that in COMPUTE state, I compute the (i,j) element of output matrix, which can be at most 4 additions and 4 multiplications in 1 clock cycle and could be the critical path. An optimization possibility is to separate them into 4 clock cycles.  Special thanks to the TA in charge for pointing out the timing (posedge/negedge) mismatch between 2 modules, which I failed to detect on my own. | | | | | | | | | |

*Scoring = (Total logic elements + total memory bit + 9\*embedded multiplier 9-bit element) (Total cycle used\*clock width)*