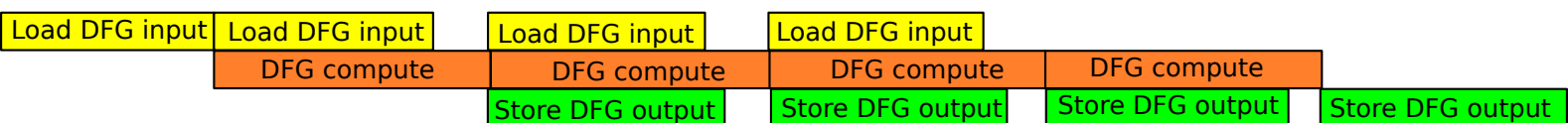


Opportunity:

1. SCGRA can have large bandwidth reading/writing data from/to the on chip buffers as long as the load time can be hidden by the computation. It will not be limited by the primitive buffer. For instance, 1kx36 bit primitive input buffer can be partitioned into 8 tiny buffers.

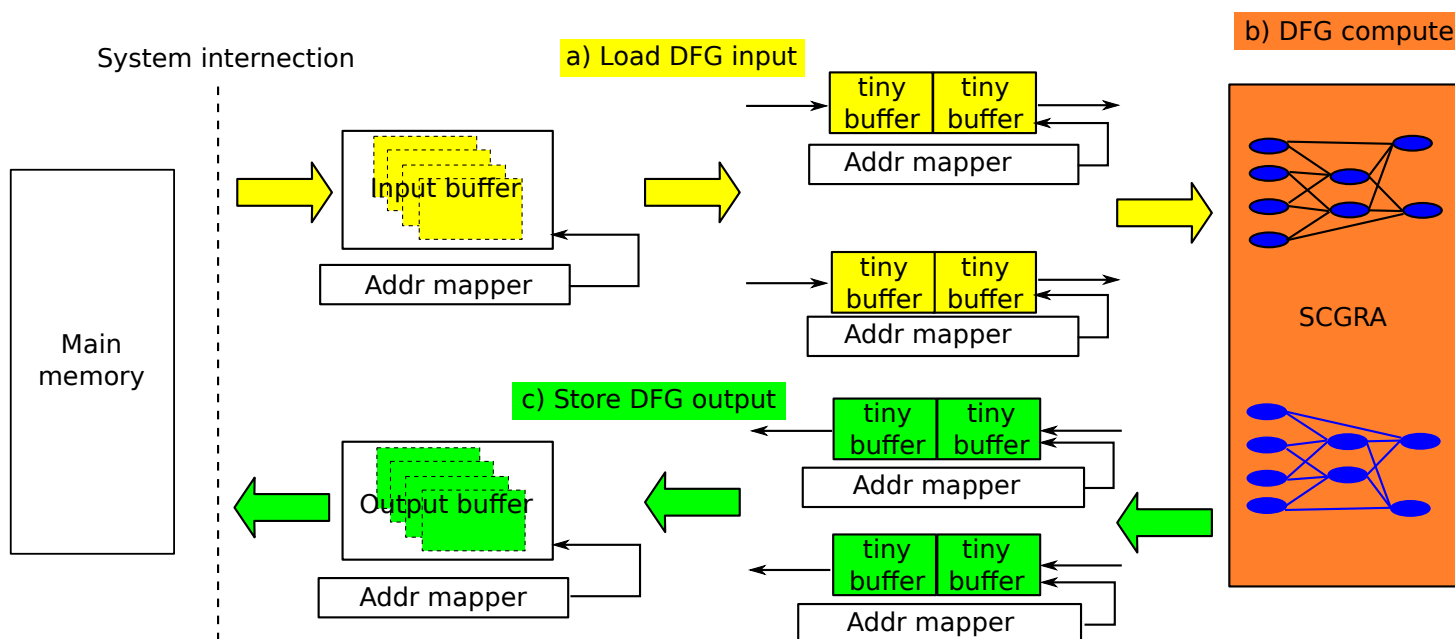
2. DFG input/output data can be placed into the tiny buffers in any order that provides better performance.

3. Can be general enough for communication between host CPU and SCGRA or two individual SCGRAs.



Physical addr: input/output data addresses in input/output buffer

Logic addr: input/output addresses of the DFG assuming a unified input/output buffer



Physical Addr to logic addr

1. Input/output buffer port number
2. How to store the io data from main memory? sequential/partitioned?

Logic addr to scheduled sequence

1. number of input/output tiny buffers
2. Depth of input/output tiny buffers
3. How to store DFG IO into these tiny buffers? sequential/partitioned/more flexible allocation?

Physical addr to scheduled sequence  
Replace tiny buffers with fifos.

Challenges:

1. How to combine it with the rest of the system customization? A bit too complex...
2. How to determine the optimal data placement for a single DFG scheduling? it depends on the scheduling and will be complex....
3. Still needs a dedicated algorithm to perform the address translation. It seems to be a typical problem, but not sure at the moment...