must match

assumes DUT a and b consumes in an order

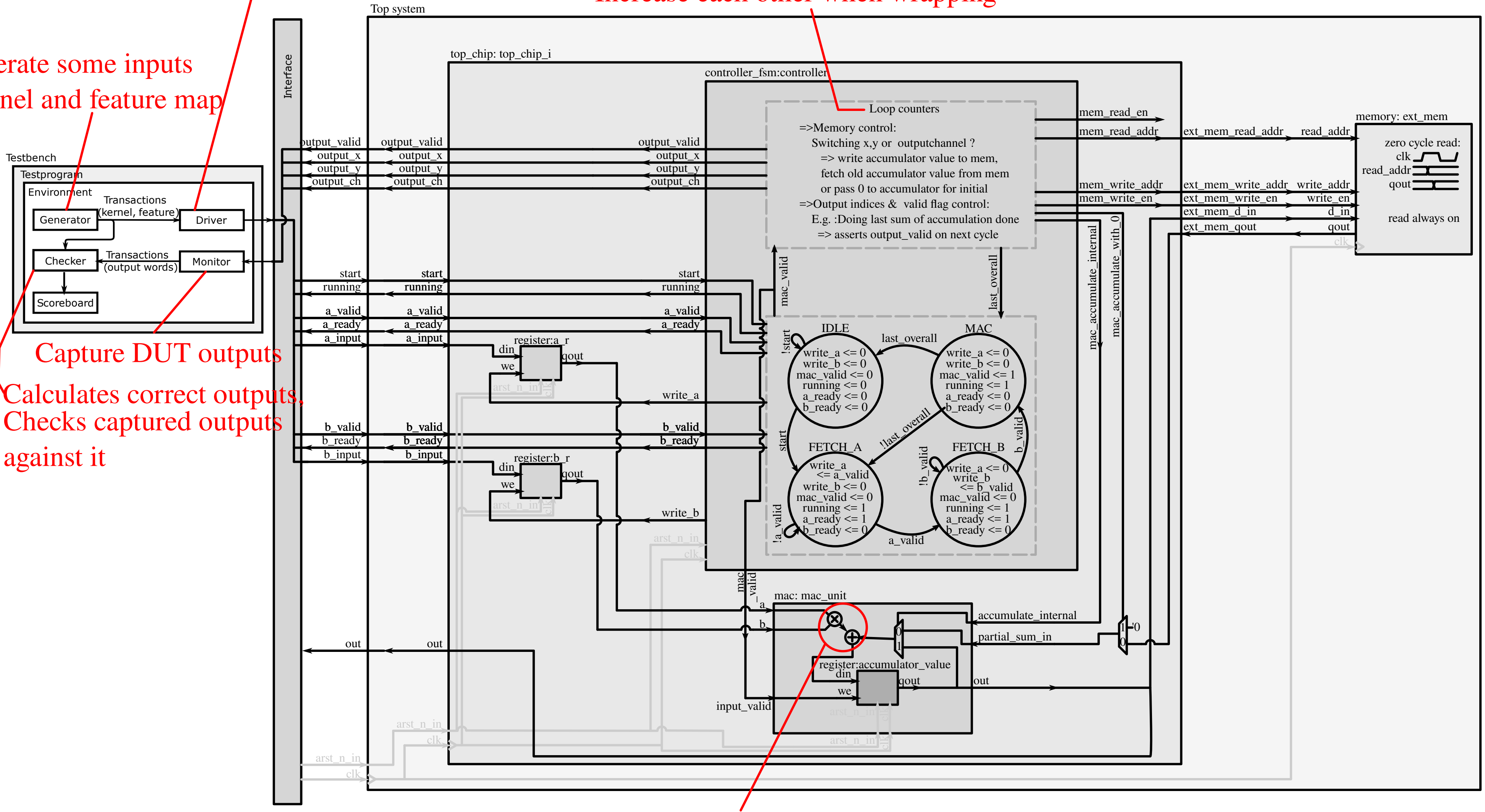
assumes tb passes a and b consumes in an order

decides loop order in tb

decides loop order in DUT

Increase each other when wrapping

Generate some inputs
=kernel and feature map



Capture DUT outputs

Calculates correct outputs,
Checks captured outputs
against it

NN = Multiply Accumulate units
=What actually needs to happen

Target = low latency = time needed to compute the CONV layer

=> Time needed to compute all the Multiply Accumulates (MACs) in the conv Layer

Latency = #cycles * time/cycle

= #cycles * (critical path length)

= #cycles * (#16b-multipliers and/or 32-bit adders chained without register in between)

(=2 in this example: 1 16-bit multiplier + 1 32-bit adder)

$$= \frac{\text{\#macs\_to\_be\_done}}{\text{average number of macs per cycle}} * \text{(\#16b-multipliers and/or 32-bit adders chained without register in between)}$$

$$= \frac{\text{\#macs\_to\_be\_done}}{\text{(nb\_mac\_units * utilization)}} * \text{(\#16b-multipliers and/or 32-bit adders chained without register in between)}$$

(=currently 1MAC unit active every third cycle = 1/3)

=> To decrease latency:

Increase number of MACs
and/or Their utilization
and/or The critical path length

will lead to changes to loops => adjust accordingly in TB and controller

Limited by:

1) bandwidth => can not infinitely pass more data in parallel to DUT

2) area => can not infinitely store data in memories/registers in DUT, nor have infinite amount of MAC units