The Multiplexers-E Project 5 - Classify

Diwan Anuj Jitendra 170070005 Soumya Chatterjee 170070010
Vabilisetti Mohan Abhyas 170260032 Arnab Jana 170100082

1 Input Output Contracts

- The valid_in bit during input from a port is continuously 1 externally, as long as the packet doesnot end.
- When the oData_rd becomes 1, the packet at the head of the FIFO (if any) is output over as many clock cycles as required by the length of the packet.
- Each packet segment given as input and output of 144 bits.
- The maximum number of hops is assumed to be 60, so path-vector has been allocated a maximum size of 480 bits.

2 Implementation

- A **32-port round-robin arbiter** chooses one of the available input ports and reads data from chosen port to registers. Arbiter sleeps till we don't set the current ack, which is set only when iData_av is non-zero at one of ports and we have gone-back to idle after operating on the current packet (if any). We use signal "valid" which says the duration during which input data is valid and can be read.
- We used a **FIFO** with din and dout of size 145 bits (including valid bits) and read and write clocks synchronous with our input clock.
- We have 2 state machines, where the second state machine is a sub part of one of the state of main state machine.

Sl no.	State	Description
1	idle	This is the state when input is not available at any of the 32 ports. Next state if iav.
2	iav	From idle, FSM enters iav if input data is available at one or more input ports. ack to the round-robin arbiter is set high, asking it to grant some available port. Next state is oav.
3	oav	From iav, FSM enters oav. This is basically a one-clock-cycle delay phase for the arbiter to grant some valid input port. Next state for FSM1 is reading and FSM2 is rinter.
4	reading	After oav it enters reading state, where packet (header+data) is being processed. From here, FSM2 comes into picture.

FSM1 for receiving available inputs to writing entire packet from a particular granted port

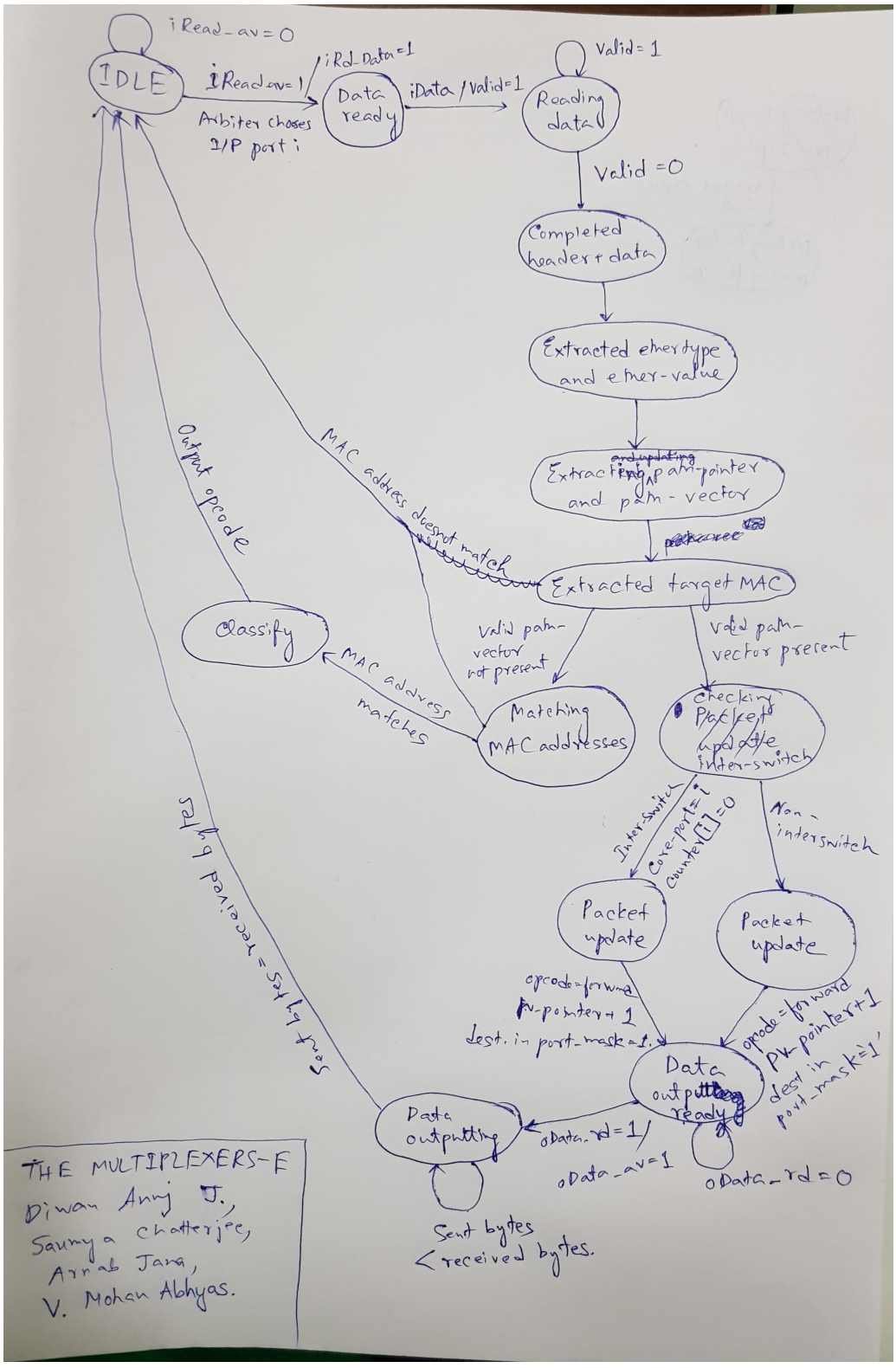
- We have 12 states in the second state machine namely "rinter" "r0", "r1", "r2", "r3", "r4", "rcont", "r0out", "r1out", "r2out", "r3out" and "rover".

Sl no.	State	Description
1	rinter	It stays in this state as long as valid is 0 even after port grant. When valid is high, next state is r0.
2	r0	Stores first subpacket in a register. Extracts ethertype, ethervalue and opcode. Next state is r1.
3	r1	Stores next subpacket in a register. Extracts no. of hops and current hop-pointer. Next state is r2.
4	r2	Stores next subpacket in a register. Next state is r3.
5	r3	Stores next subpacket in a register. Next state is r4.
6	r4	Stores next subpacket in a register. Finish extracting path-vector Extracts target-MAC. It processes the path vector and MAC here. Next state ir rcont.
7	rcont	Store the current packet in a register. Shift packet0 by packet1, packet1 by packet2 and so on. At this stage, packet0 is being pushed in FIFO Remains in this state as long as valid_in is high. If valid become 0, next state is r0out.
8	r0out	This is the stage when reading packet is complete. The first buffered packet is written into FIFO. Next state is r1out.
9	r1out	The second buffered packet is written into FIFO. Next state is r2out.
10	r2out	The third buffered packet is written into FIFO. Next state is r3out.
11	r3out	The last buffered packet is written into FIFO. Next state is rout.
12	rout	At this state, the entire writing is complete. Next state for FSM1 is idle.

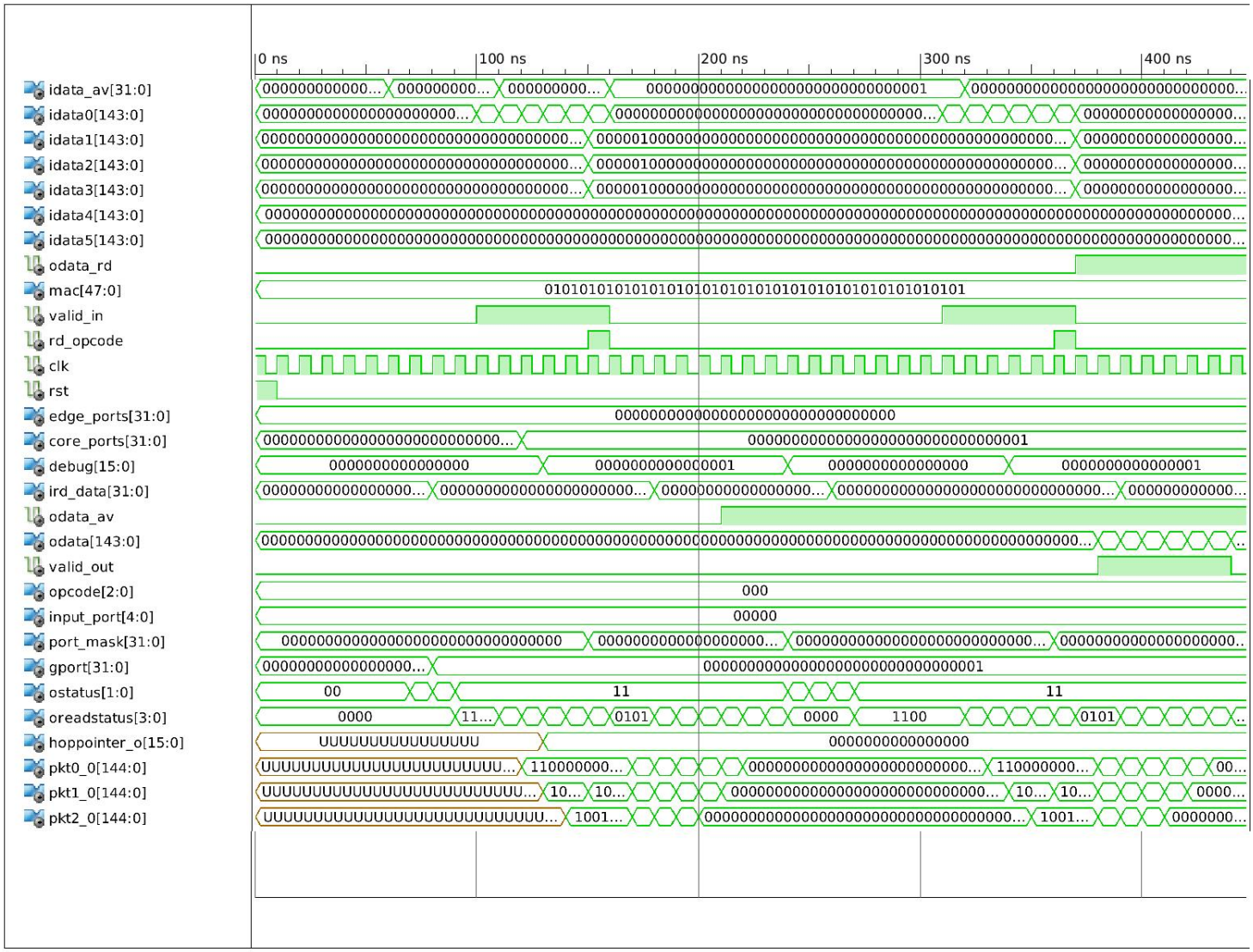
FSM2 for reading data and writing to queue

- After processing the header, if valid path-vector present, we **increment hop-pointer** and **mark opcode as forward** in the packet.
- Otherwise, if **target MAC does not match** with device's MAC, the packet is dropped.
- If it matches, we **write the packet into the FIFO**.
- The parallel process outputs the data after oData_rd is made 1. We use valid bit for each chunk, and use an invalid bit to denote a separator in the FIFO to distinguish between chunks of two different packets.
- We maintain **counters for each of 32 ports**, which are reset to count a tick of 1sec, once the first inter-switch packet arrives and set the port as core-port.
- For subsequent packets, if the timer times out, we mark the port as edge port and if we receive an inter-switch packet, we mark it as core-port and reset the timer.

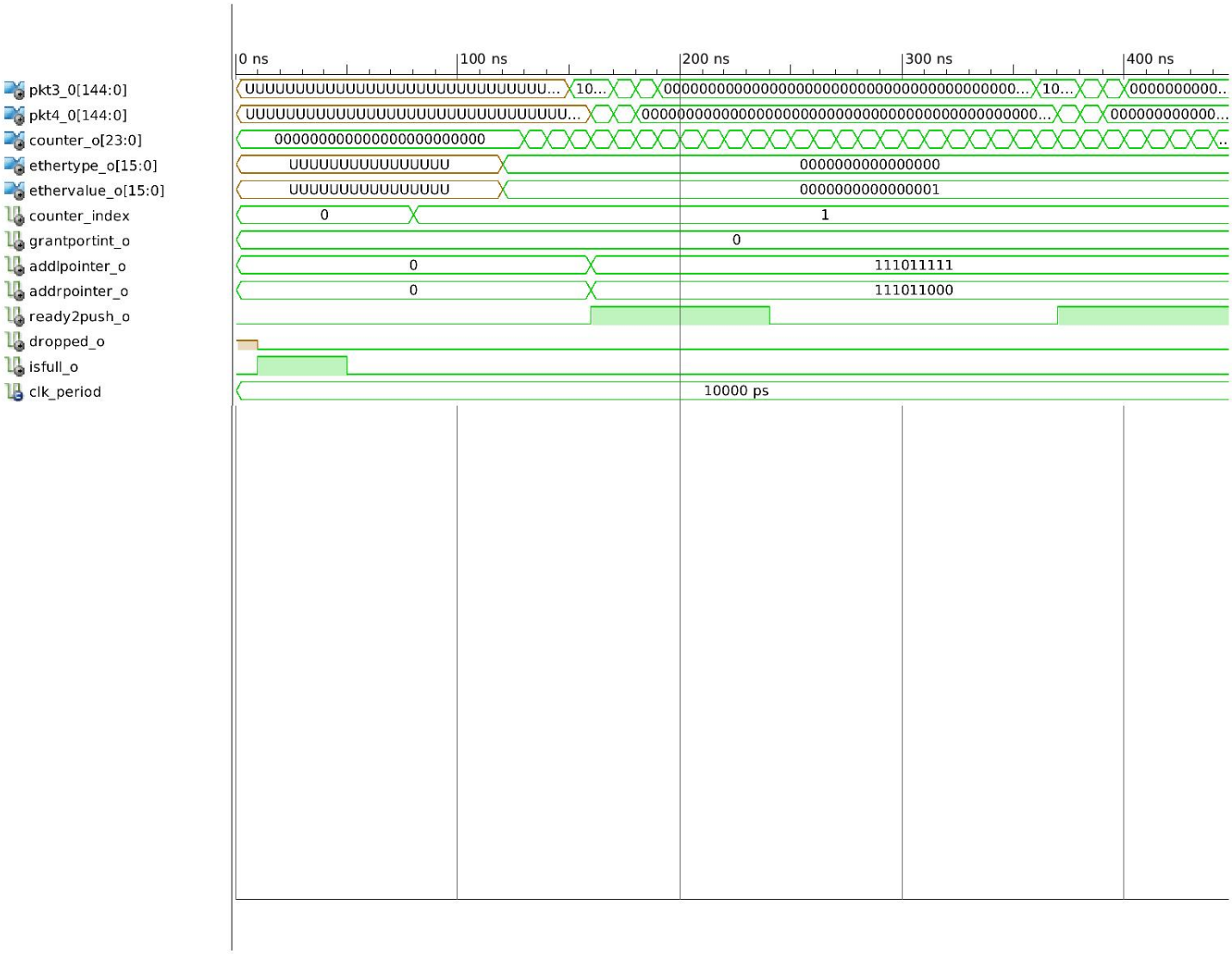
3 Finite State Machine



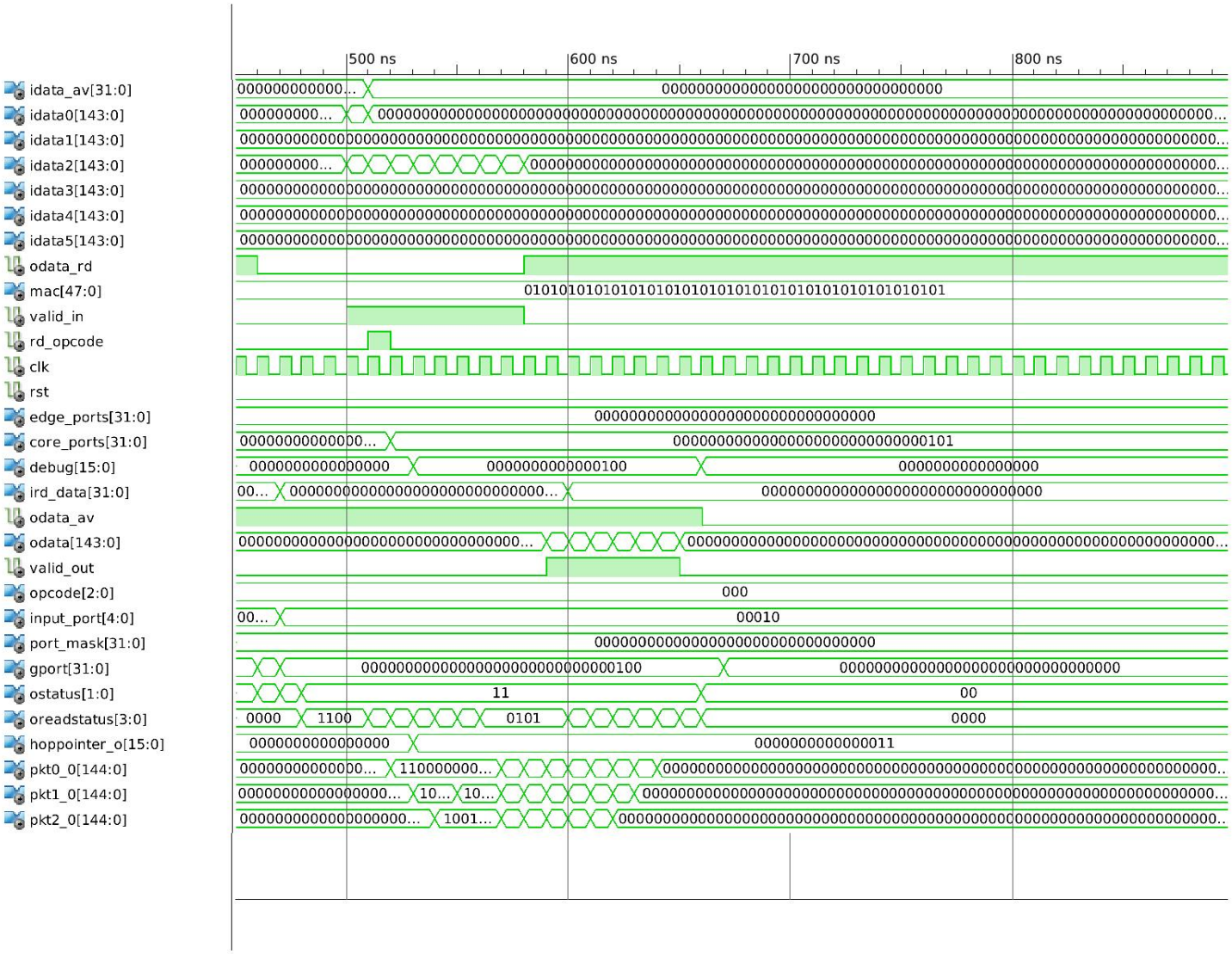
FSM Classifier



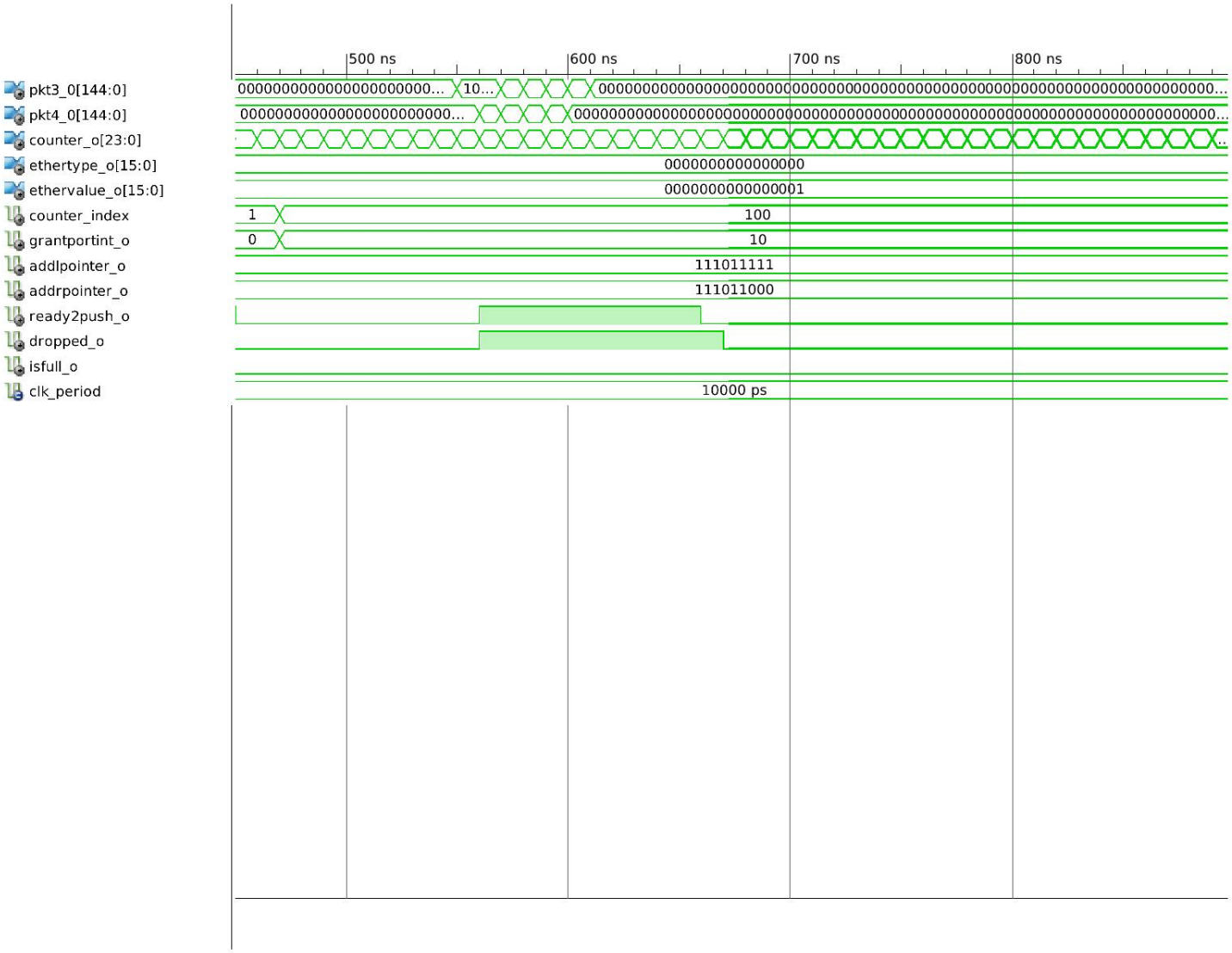
Timing diagram: Page 1



Timing diagram: Page 2



T
iming diagram: Page 3



Timing diagram: Page 4