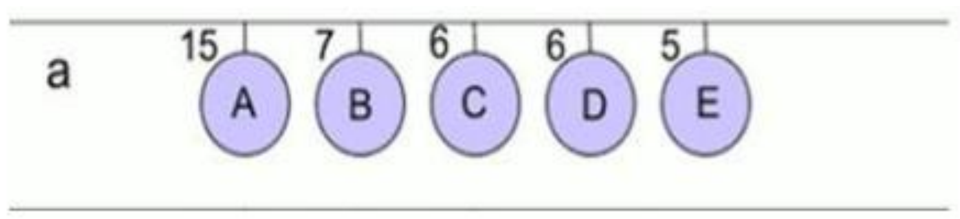
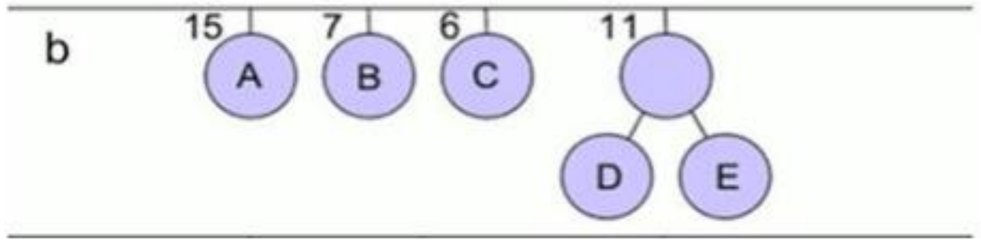
Lab06
Huffman Code Operation

Analysis

Combine two nodes in each cycle

- For every case, it requires 7 latencies.
- Critical path: a sorting network & a 5 bits adder to calculate weight.

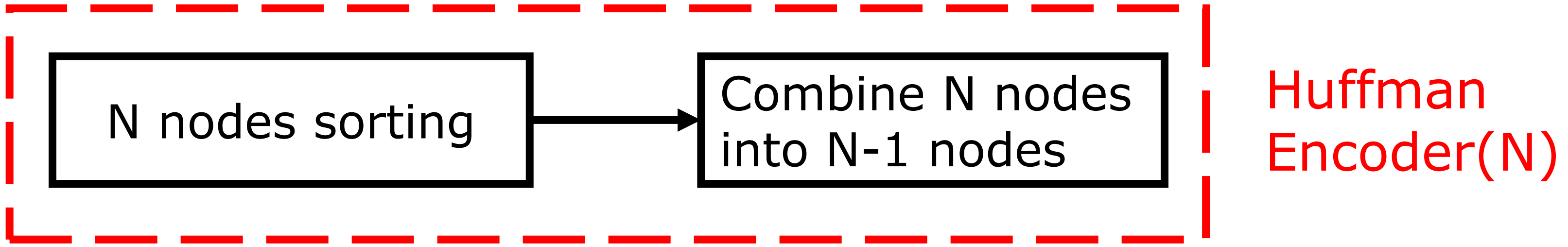


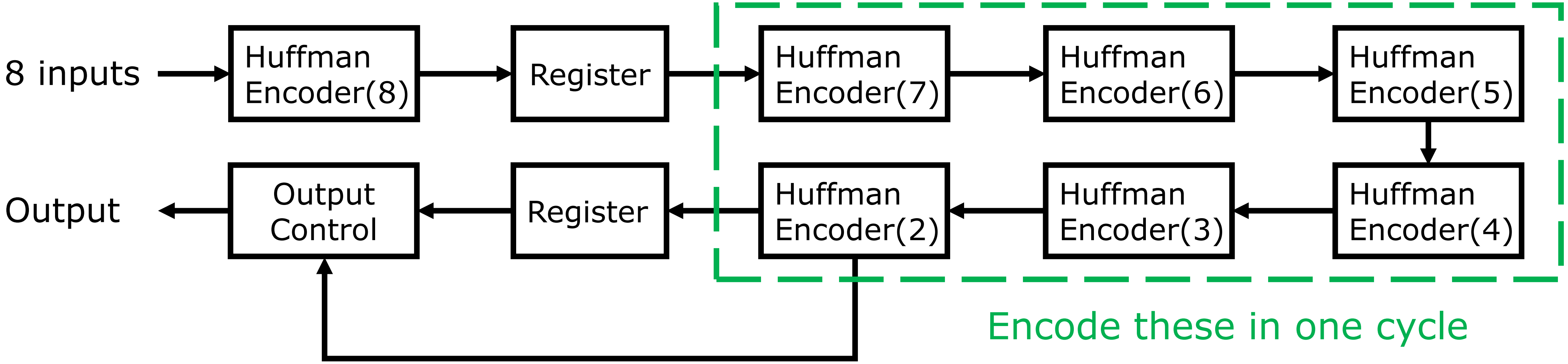


Strategy

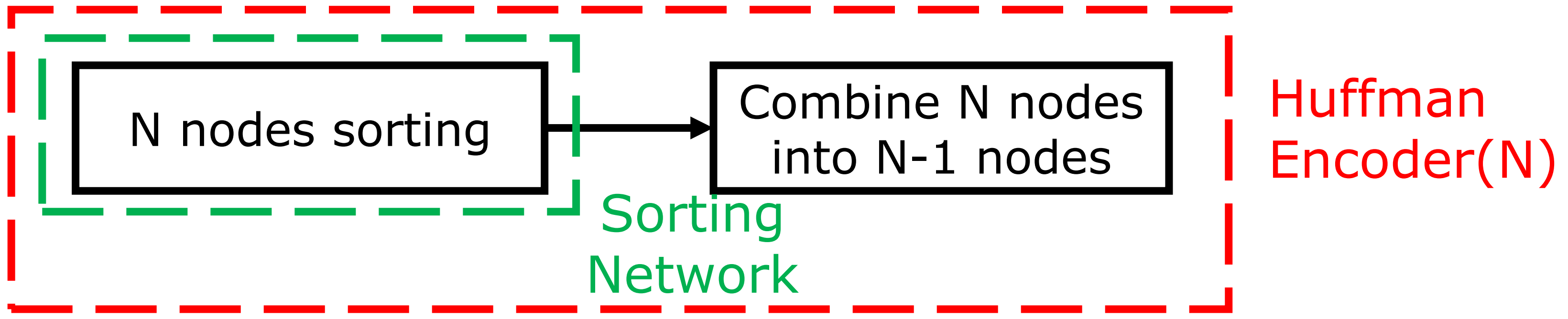
- Reduce the latency for each case to 1.
- A limitation on reducing cycle time.
- The trade-off is still worthwhile due to the 1/7 latency.

Architecture





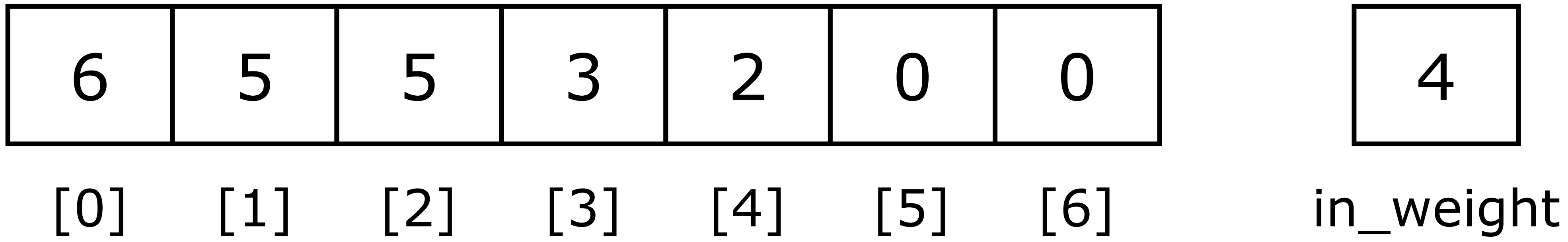
Sorting Network

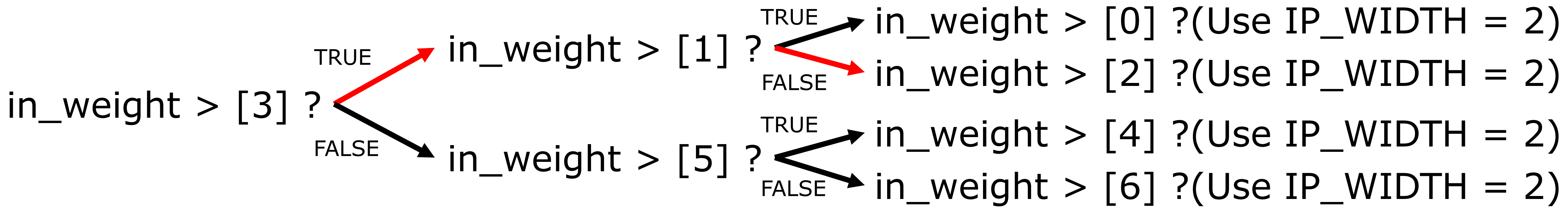


- Use binary search to find the position where the element should be inserted.
- Use SORT_IP with IP_WIDTH = 2.
- Use bubble sort to design SORT_IP

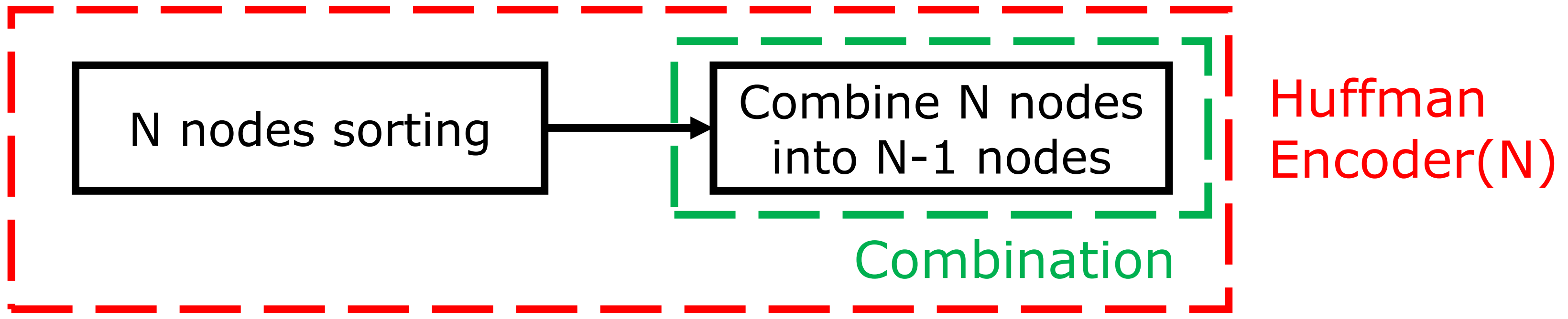
Sorting Network

Ex. For 8 nodes sorting




in_weight > [3] ?
TRUE
FALSE
in_weight > [1] ?
in_weight > [5] ?
TRUE
FALSE
TRUE
FALSE
in_weight > [0] ?(Use IP_WIDTH = 2)
in_weight > [2] ?(Use IP_WIDTH = 2)
in_weight > [4] ?(Use IP_WIDTH = 2)
in_weight > [6] ?(Use IP_WIDTH = 2)

Combination



For every node

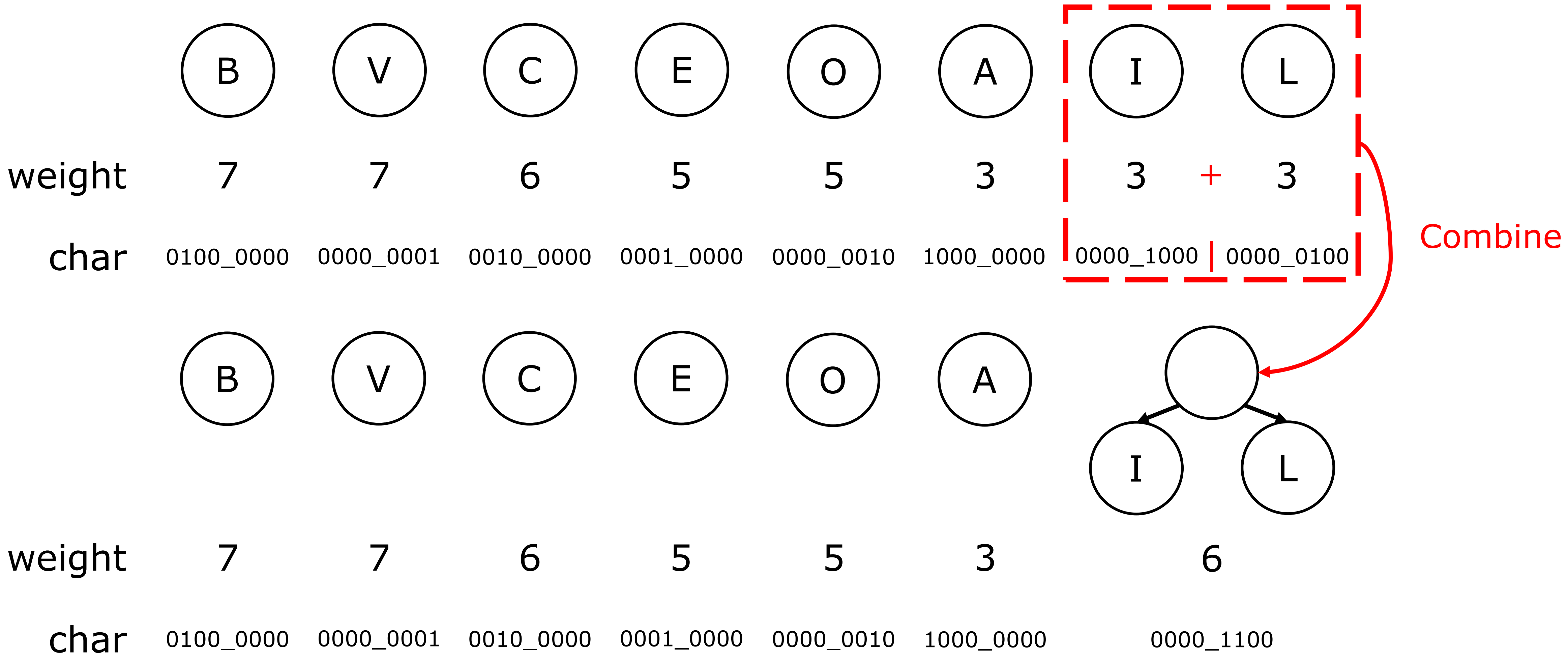
- Use signal ‘weight’ to store the weight.
- Use signal ‘char’ to store which characters are included.

For every character

- Use signal ‘length’ to store the Huffman code length.
- Use signal ‘code’ to store the Huffman code.

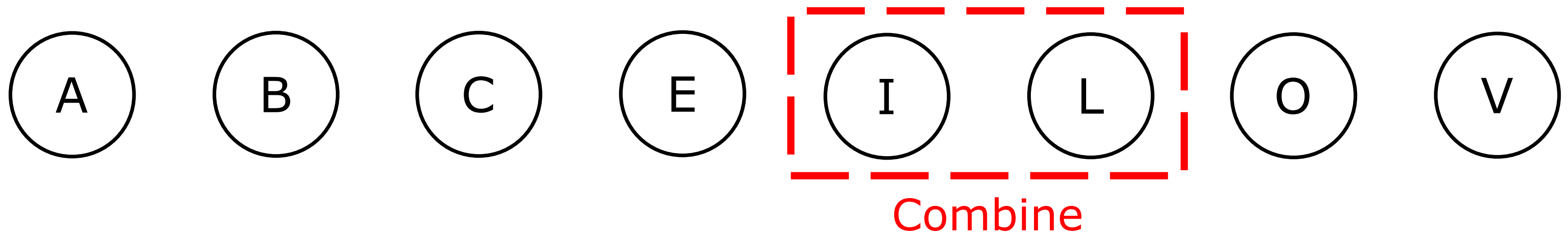
Combination

Ex. Combine 8 nodes into 7 nodes (for every node)

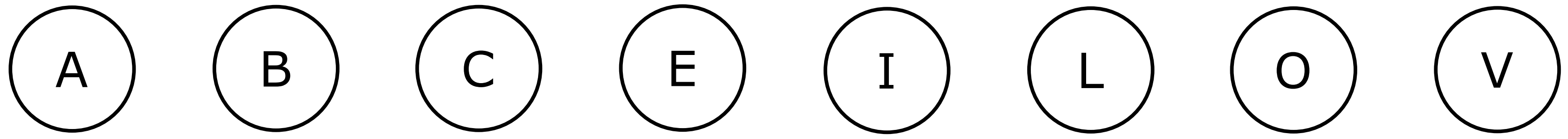


Combination

Ex. Combine 8 nodes into 7 nodes (for every character)



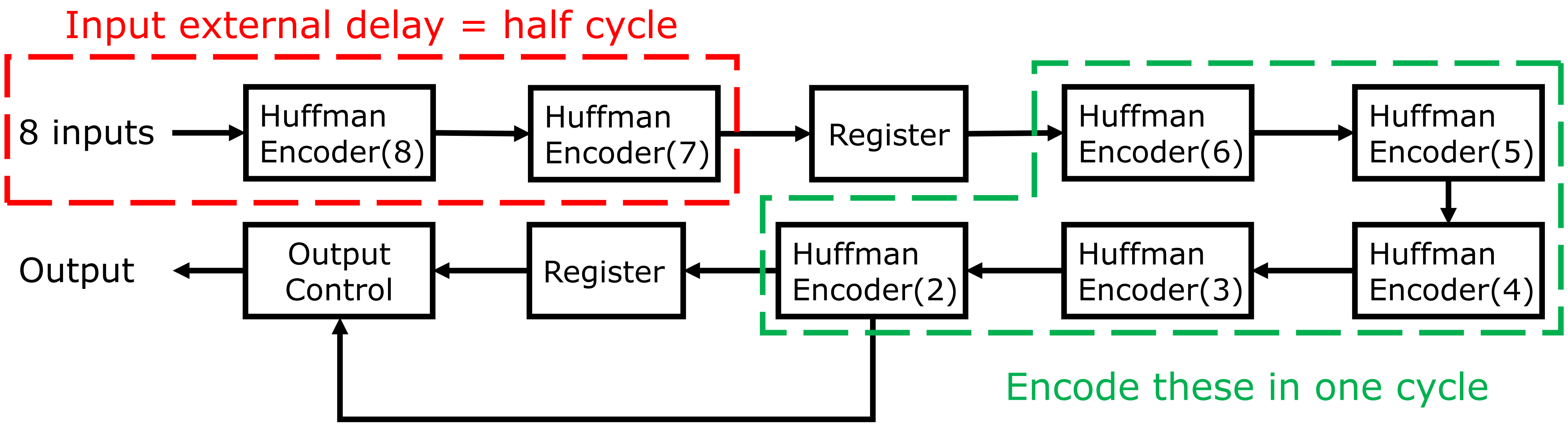
	A	B	C	E	I	L	O	V
length	0	0	0	0	0	0	0	0
code	000_0000	000_0000	000_0000	000_0000	000_0000	000_0000	000_0000	000_0000



	A	B	C	E	I	L	O	V
length	0	0	0	0	1	1	0	0
code	000_0000	000_0000	000_0000	000_0000	000_0000	000_0001	000_0000	000_0000

Input External Delay

At first



- Cannot reduce cycle time due to input external delay.
- Solution -> Move Huffman Encoder(7) to the next cycle.