CDA 4253 FPGA System Design
PicoBlaze Interface

Hao Zheng
Comp Sci & Eng
U of South Florida

Required Reading

- P. Chu, *FPGA Prototyping by VHDL Examples*

  Chapter 16, PicoBlaze I/O Interface

  Chapter 17, PicoBlaze Interrupt Interface

  Xilinx PicoBlaze User Guide (UG 129)
  Chapter 4 & 6

Input Instructions

INPUT sX, KK PORT_ID <= KK direct

sX <= IN_PORT

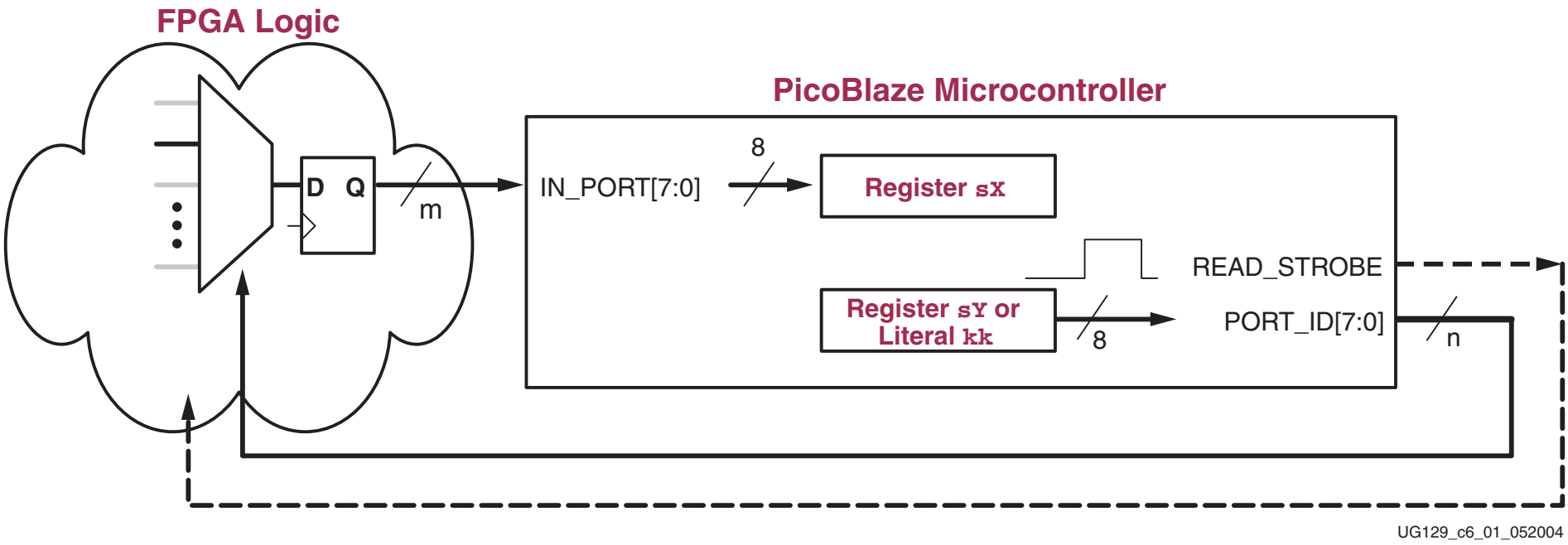
kk is the address of a peripheral.

INPUT sX, (sY) PORT_ID <= sY indirect

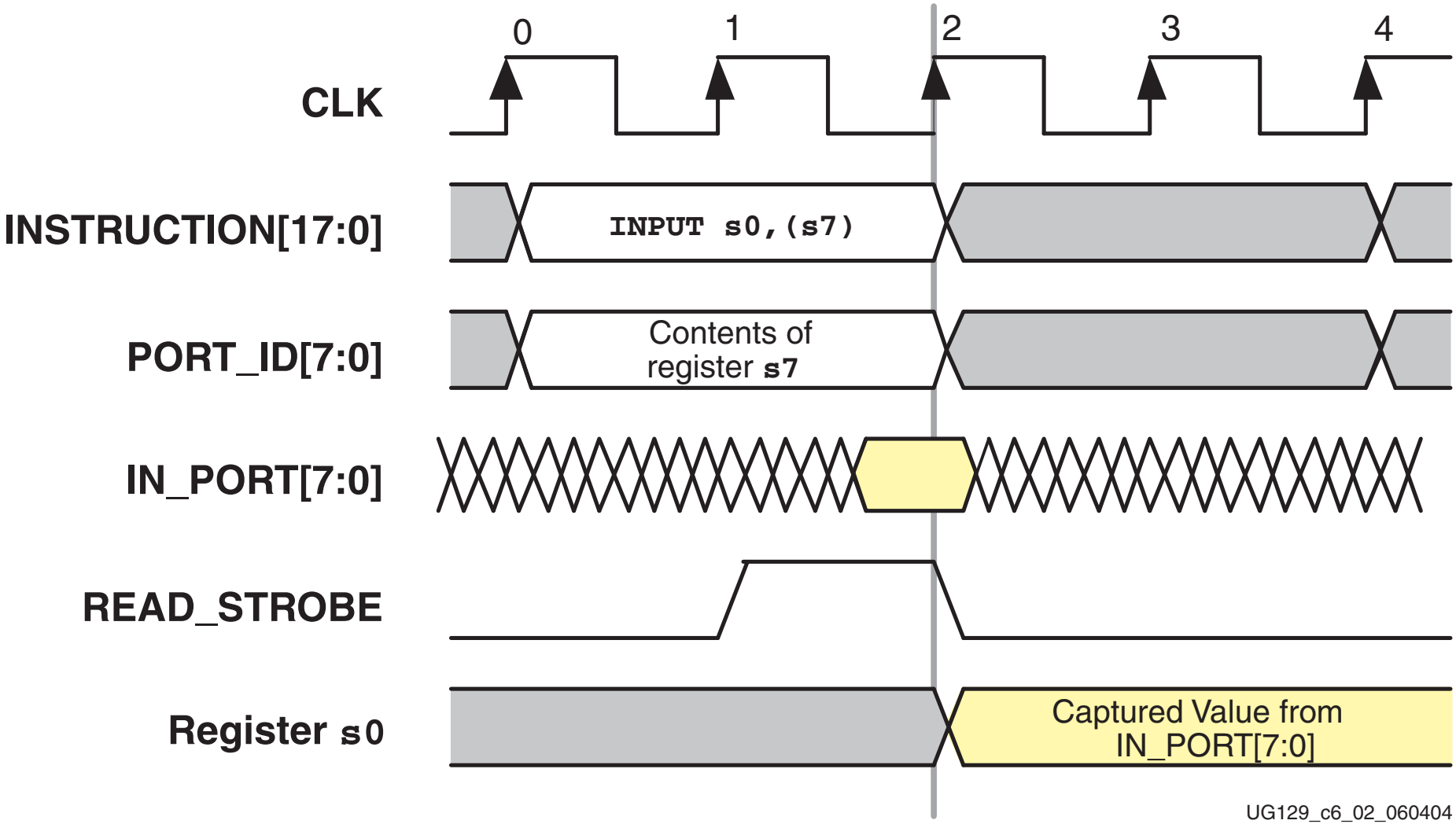
sX <= IN_PORT

sY stores address of a peripheral.

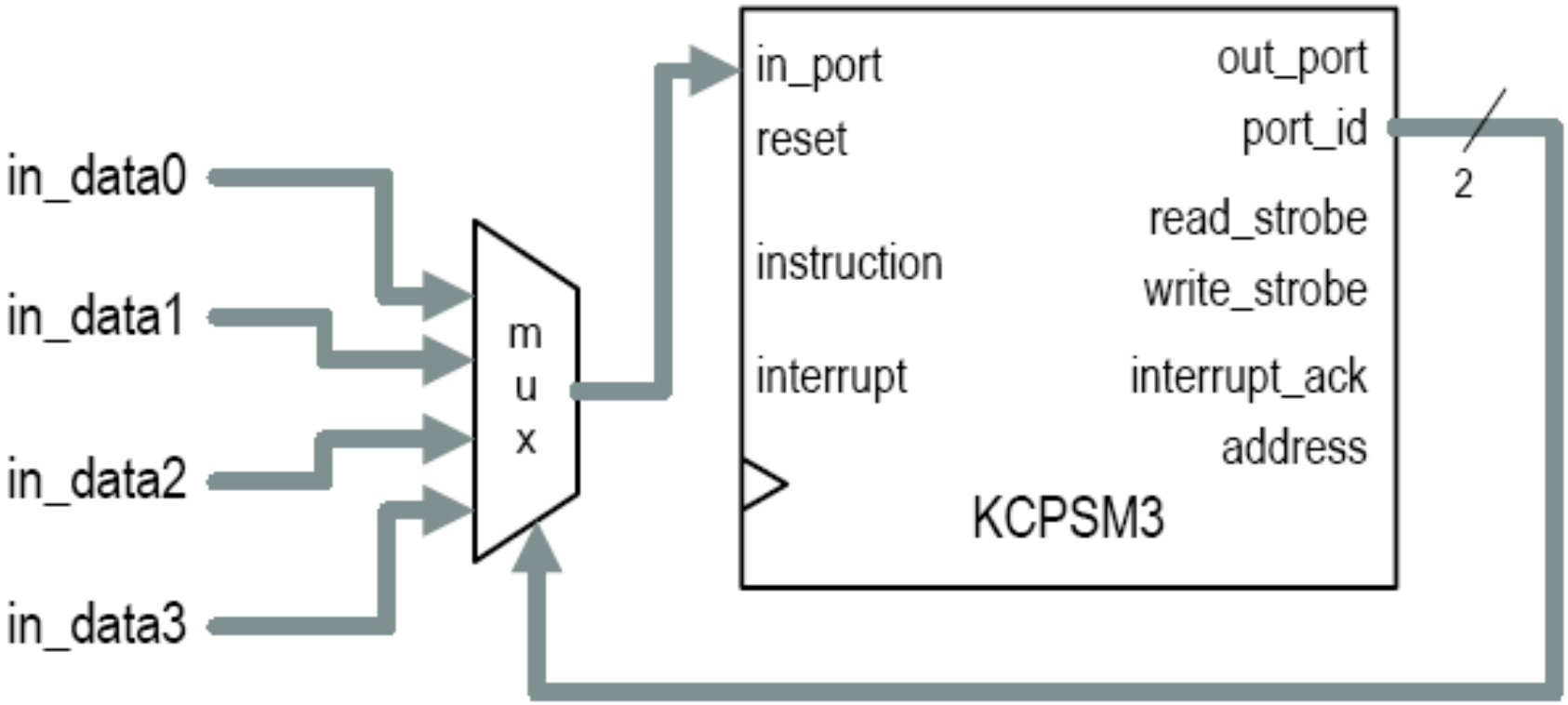
Input Operation and FPGA Interface

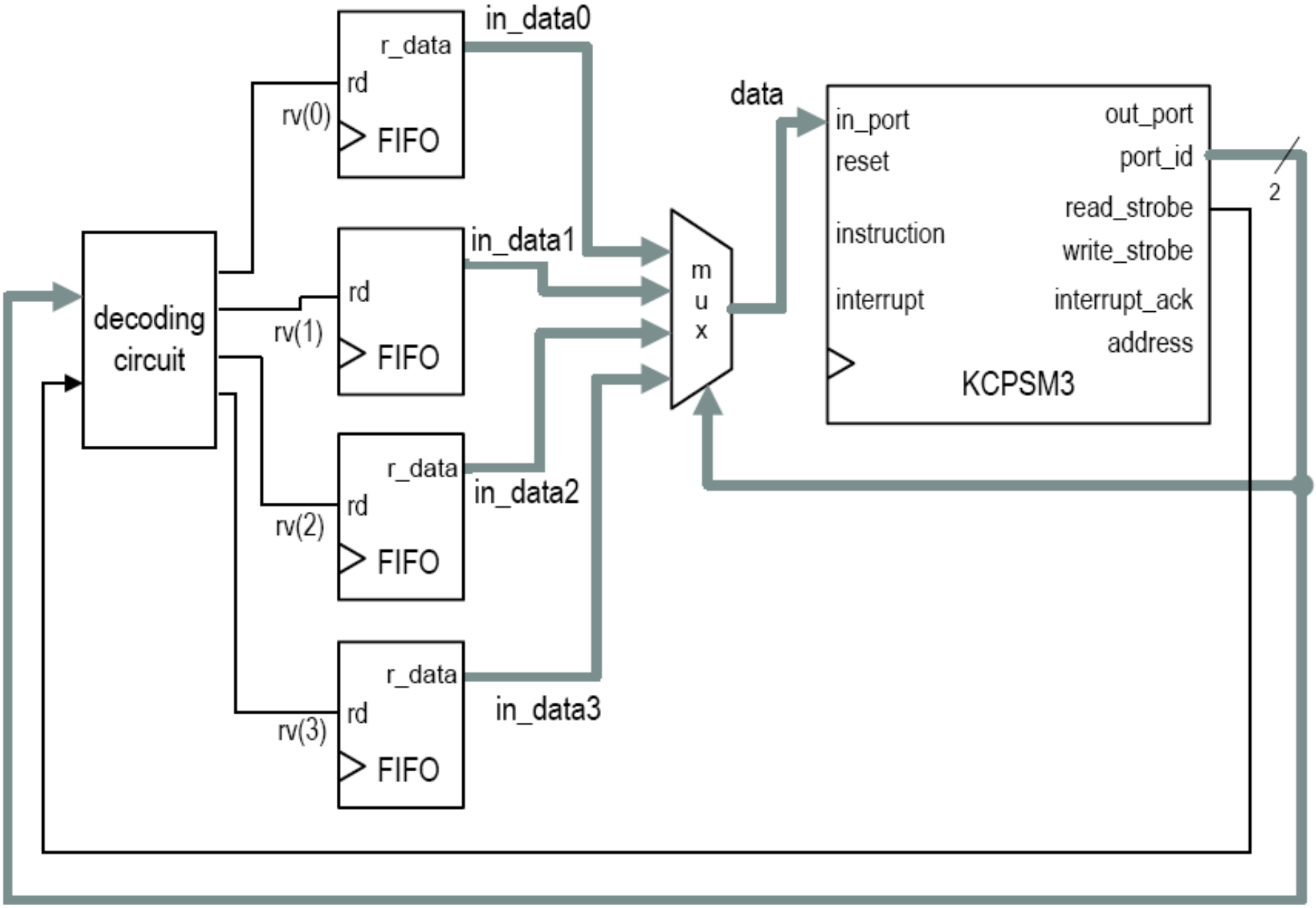


Input Instructions: Timing Diagram



Four Continuous-Access Ports



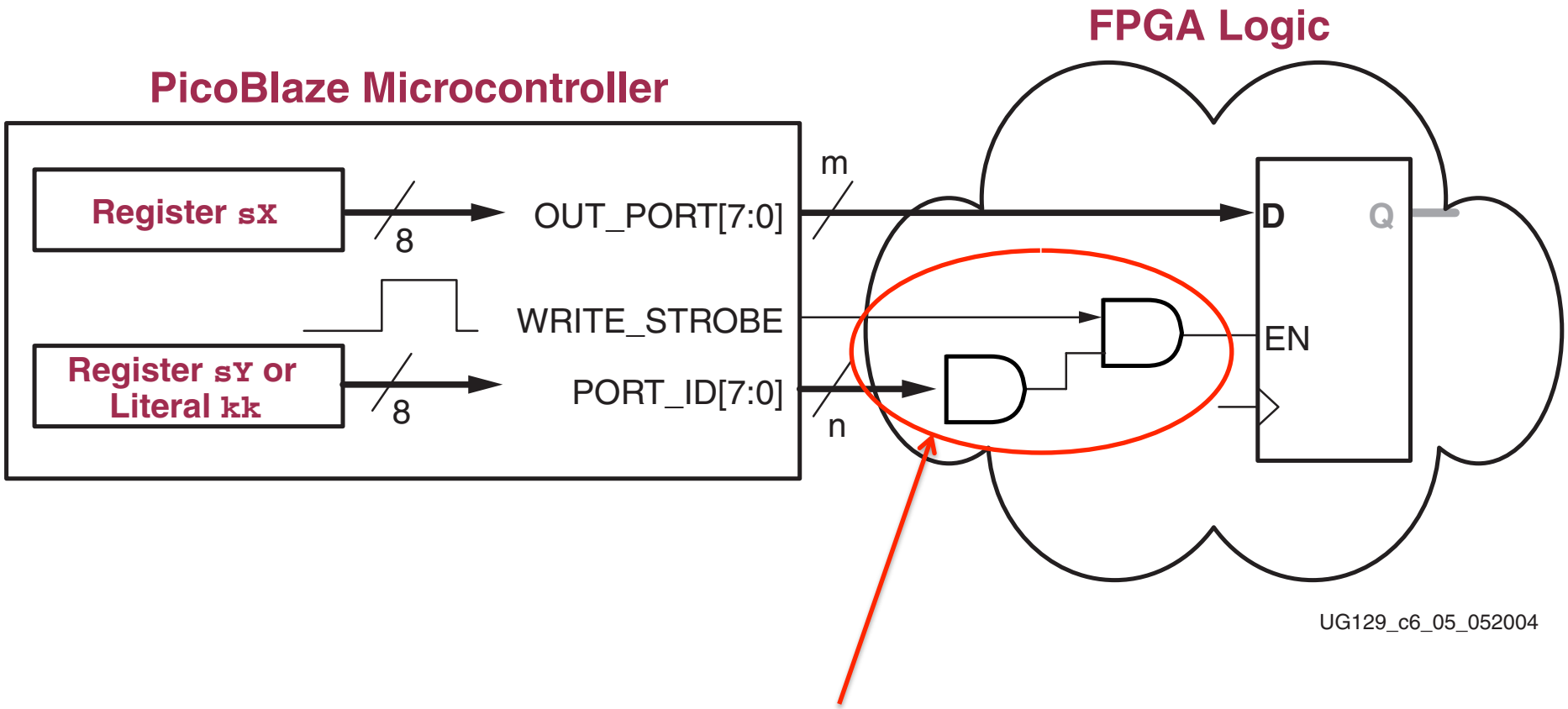

in_data0
r_data
rd
rv(0)
FIFO
data
in_port
reset
instruction
interrupt
out_port
port_id
read_strobe
write_strobe
interrupt_ack
address
KCPSM3
2
in_data1
rd
rv(1)
FIFO
m
u
x
decoding
circuit
r_data
rd
rv(2)
FIFO
in_data2
r_data
rd
rv(3)
FIFO
in_data3

Output Instructions

OUTPUT sX, KK	PORT_ID <= KK OUT_PORT <= sX	direct
OUTPUT sX, (sY)	PORT_ID <= sY OUTPUT <= sX	indirect

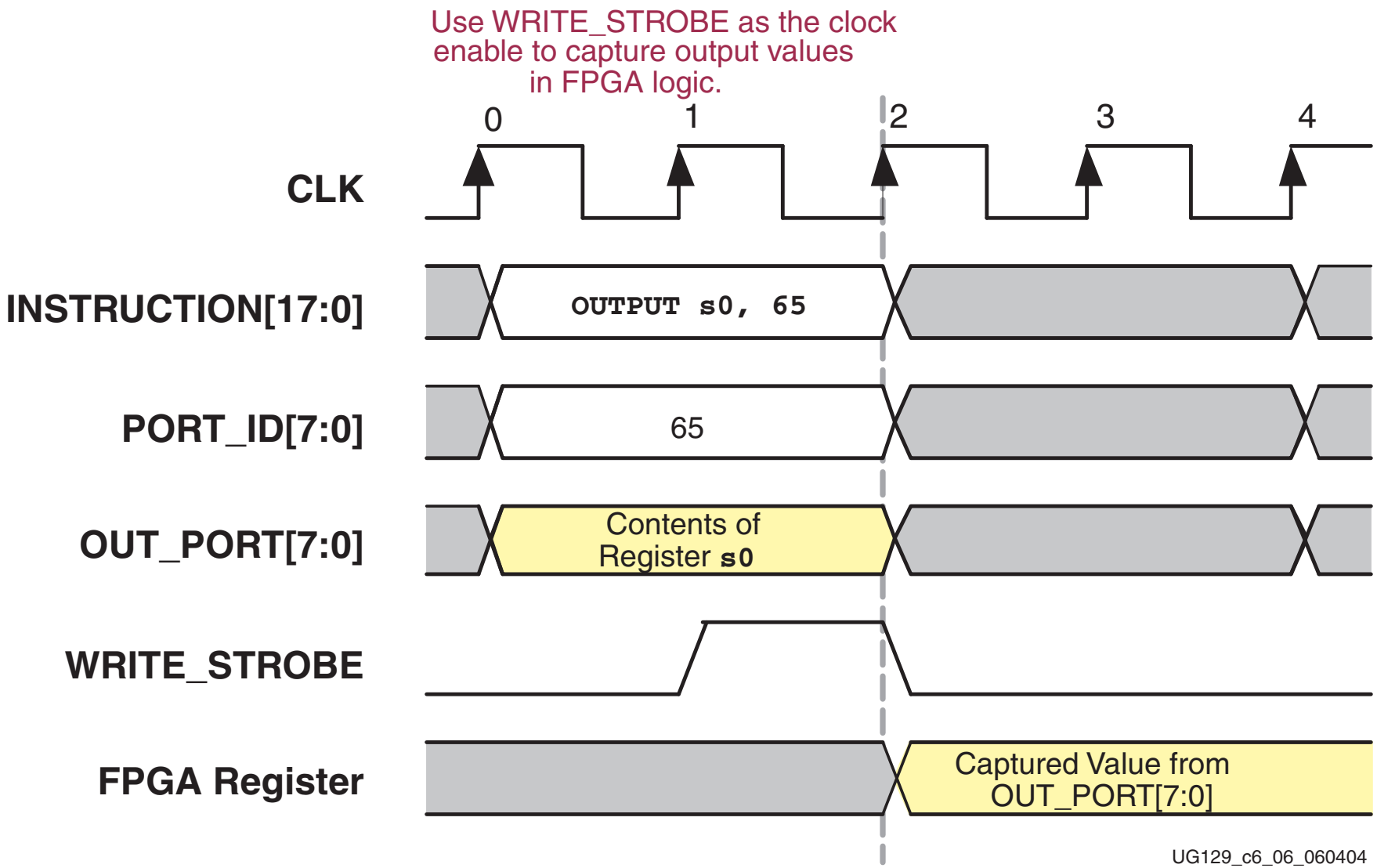
- write_strobe is asserted in the 2nd cycle of an **output** instruction.
- Used to notify target the validaty of the data on out_port

Output Interfacing

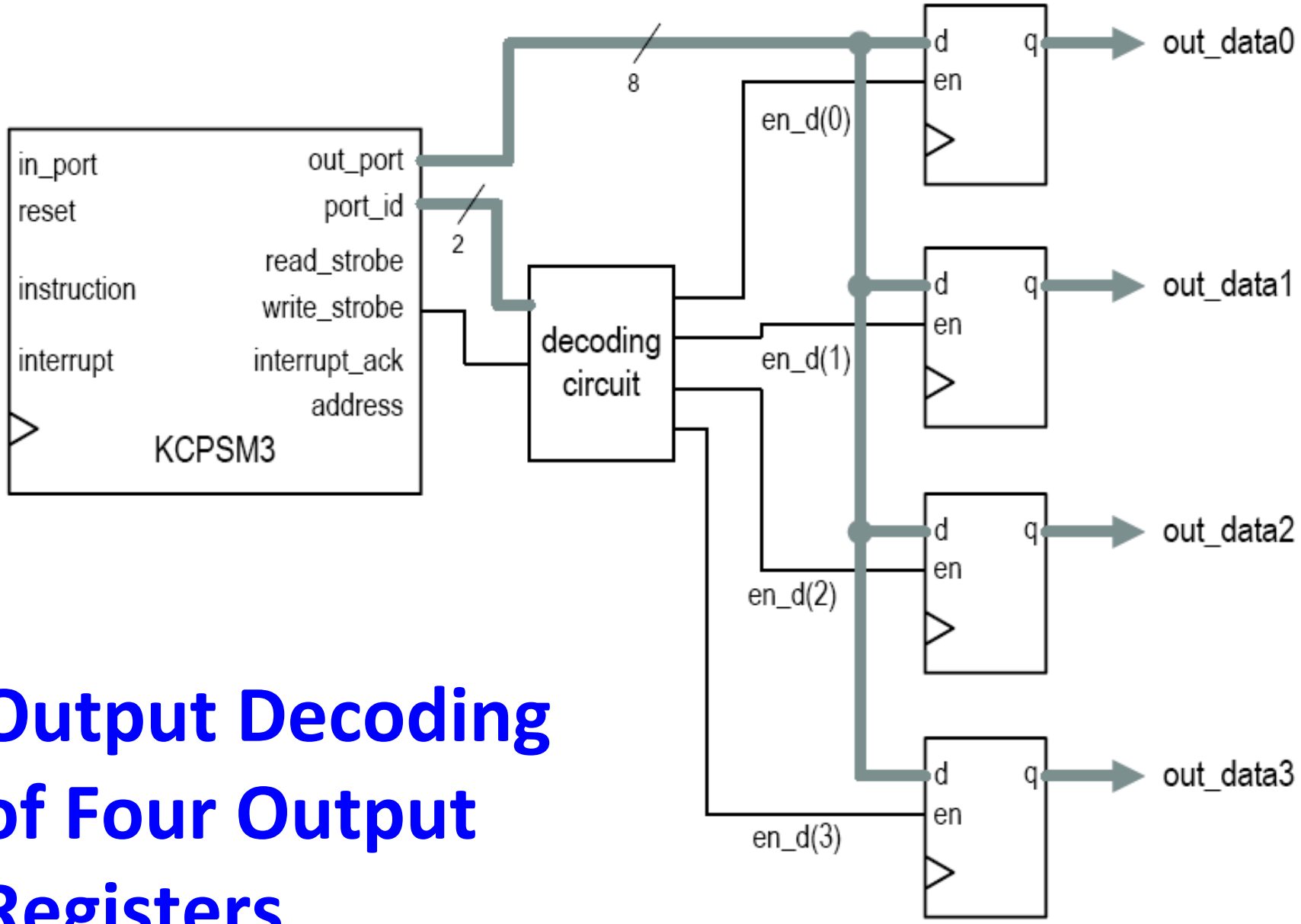


Decode Logic

Output Operation and FPGA Interface



Output Decoding of Four Output Registers



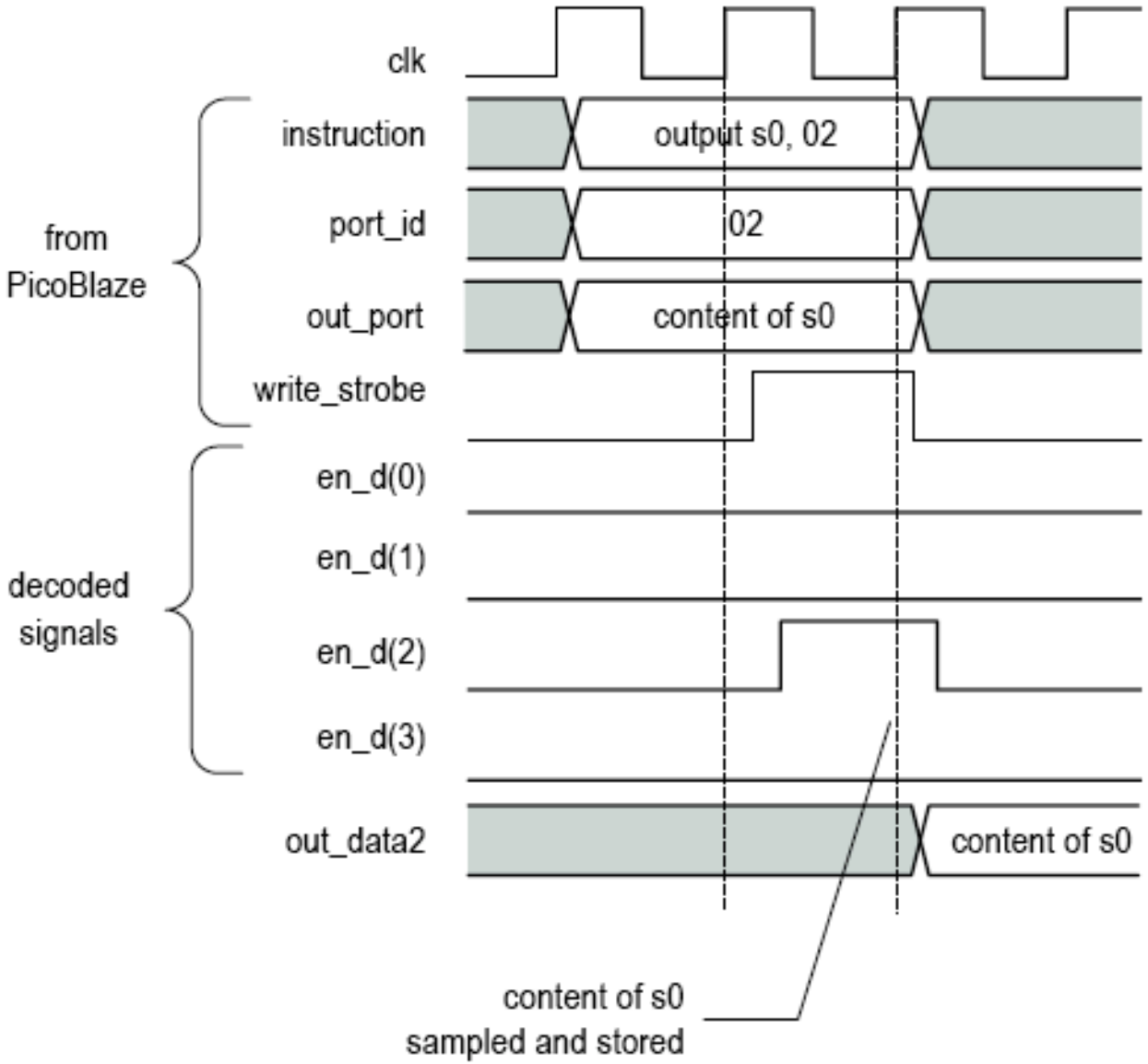
Decoder Logic

input			output
write_strobe	port_id(1)	port_id(0)	en_d
0	–	–	0000
1	0	0	0001
1	0	1	0010
1	1	0	0100
1	1	1	1000

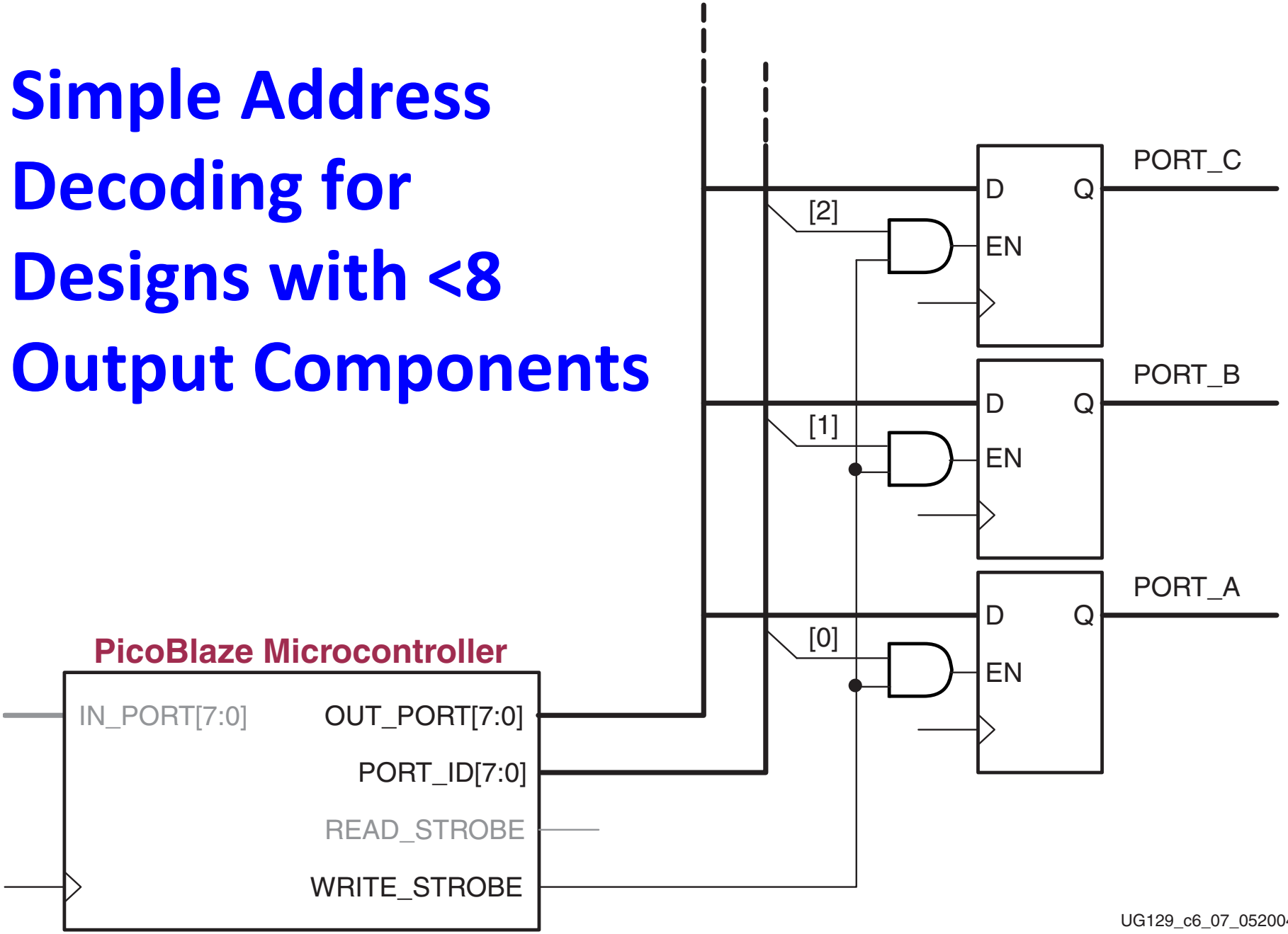
The decoder logic can be saved by using one-hot codes for **port_id** if the output ports are smaller than 8.

Timing Diagram of an Output Instruction

output s0, 02

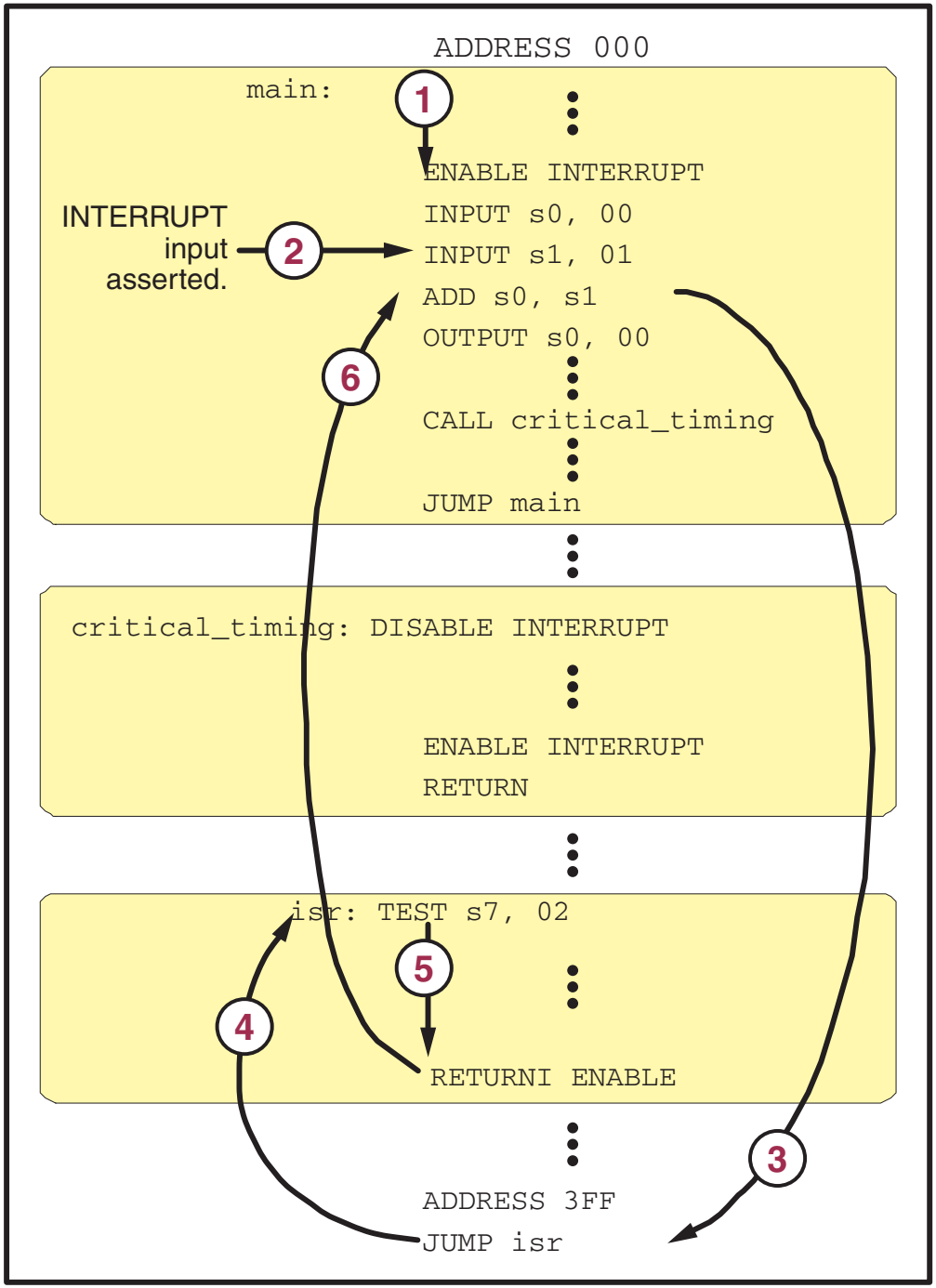


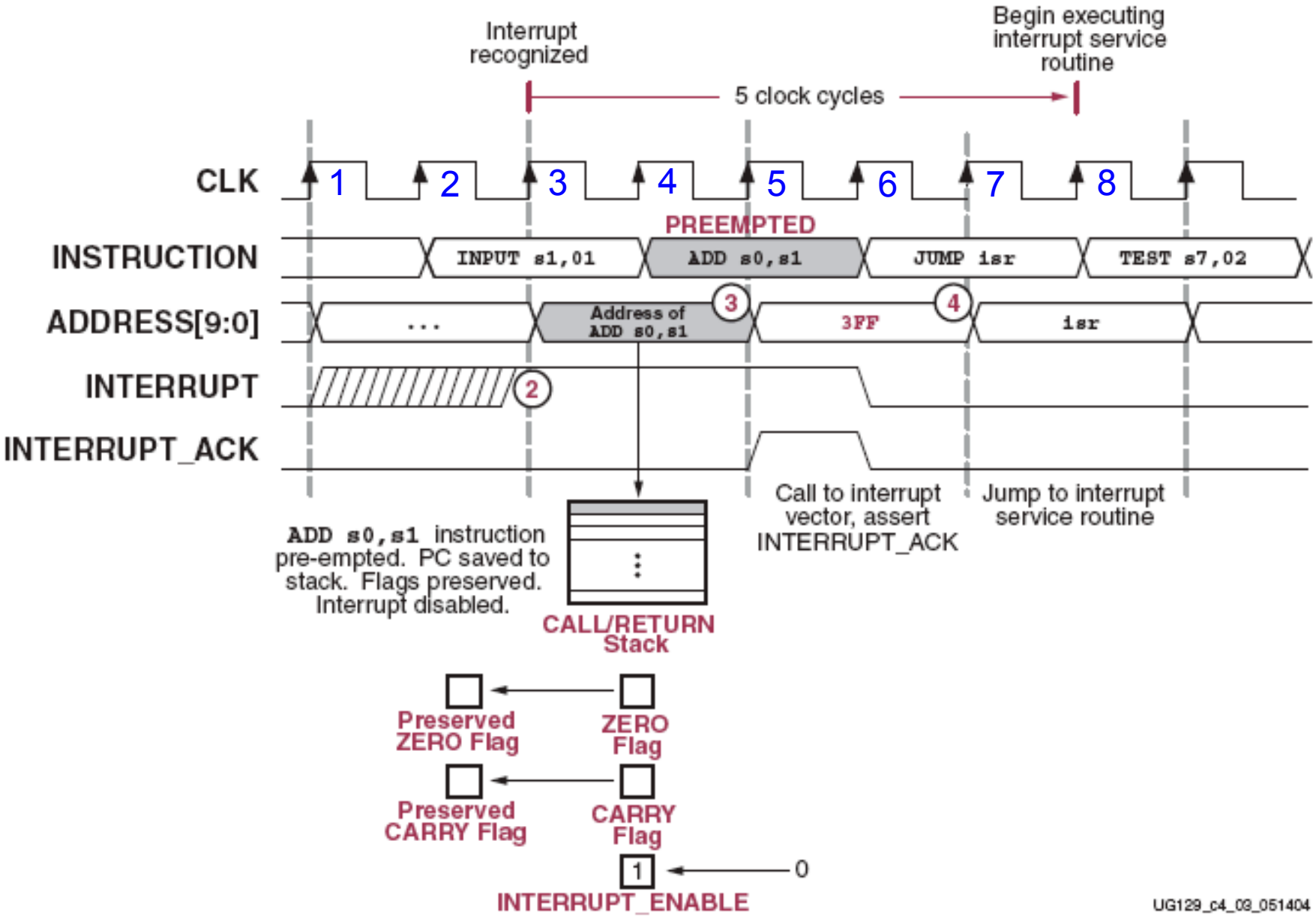
Simple Address Decoding for Designs with <8 Output Components



Interrupt Flow

1. Enabled interrupts
2. An interrupt occurs
3. Execute **call 3FF**
4. Execute **jump isr**
5. Execute interrupt service routine (ISR) at **isr**
6. execute **returni enable** at the end of ISR, and resume the normal operation




Interrupt recognized
Begin executing interrupt service routine
5 clock cycles
CLK
1
2
3
4
5
6
7
8
PREEMPTED
INSTRUCTION
INPUT s1,01
ADD s0,s1
JUMP isr
TEST s7,02
ADDRESS[9:0]
...
Address of ADD s0,s1
3FF
isr
3
4
INTERRUPT
2
INTERRUPT_ACK
ADD s0,s1 instruction pre-empted. PC saved to stack. Flags preserved. Interrupt disabled.
CALL/RETURN Stack
Call to interrupt vector, assert INTERRUPT_ACK
Jump to interrupt service routine
Preserved ZERO Flag
ZERO Flag
Preserved CARRY Flag
CARRY Flag
1
0
INTERRUPT_ENABLE
UG129_c4_09_051404

Interrupt Related Instructions

RETURNI ENABLE

PC <= STACK[TOS] ; TOS <= TOS – 1;

I <= 1; C<= PRESERVED C; Z<= PRESERVED Z

RETURNI DISABLE

PC <= STACK[TOS] ; TOS <= TOS – 1;

I <= 0; C<= PRESERVED C; Z<= PRESERVED Z

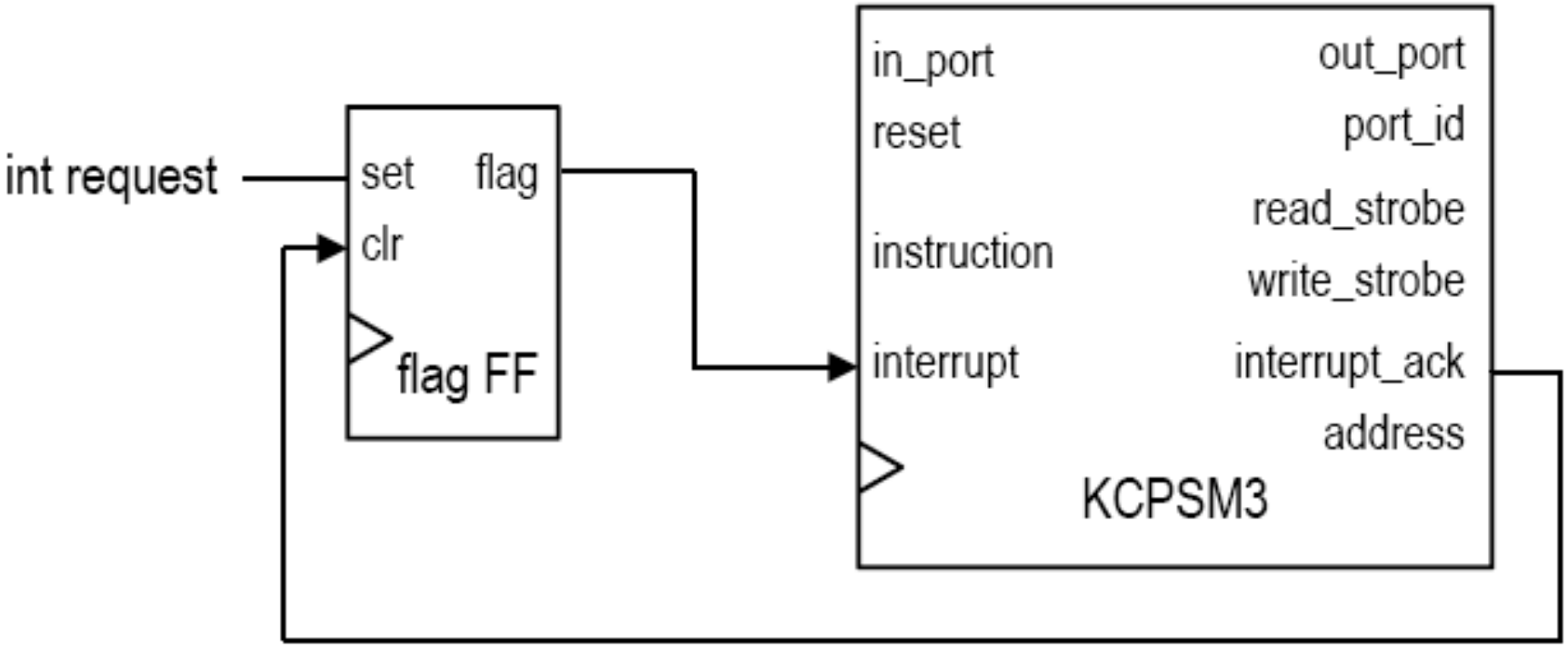
ENABLE INTERRUPT

I <=1;

DISABLE INTERRUPT

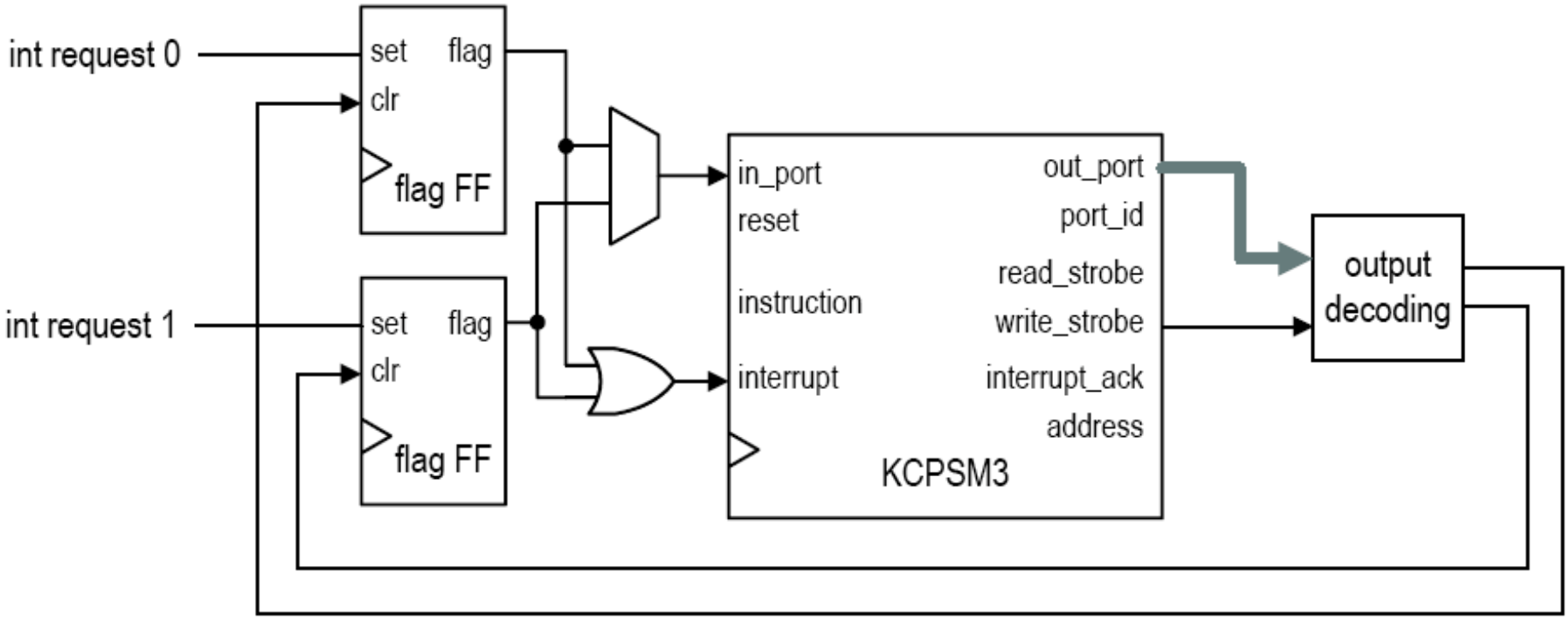
I <=0;

Interrupt Interface with a Single Event



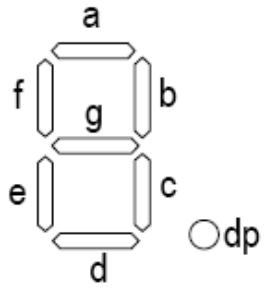
interrupt should hold high until **interrupt_ack** is asserted.

Interrupt Interface with Two Requests

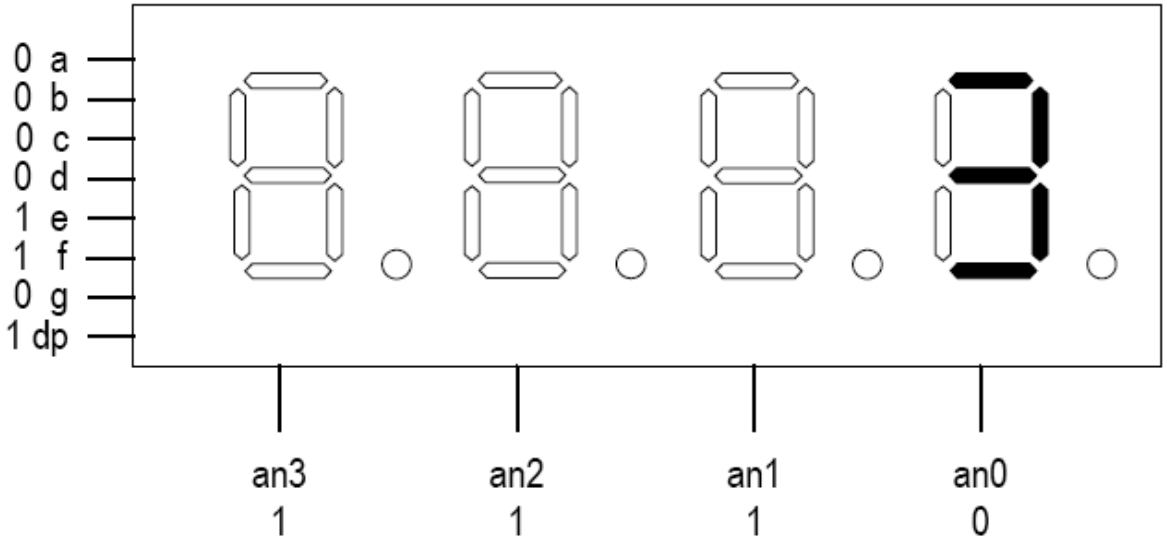


ISR reads **in_port[1:0],** decides which request should be served, and generate correct signal to clear the corresponding interrupt request FF.

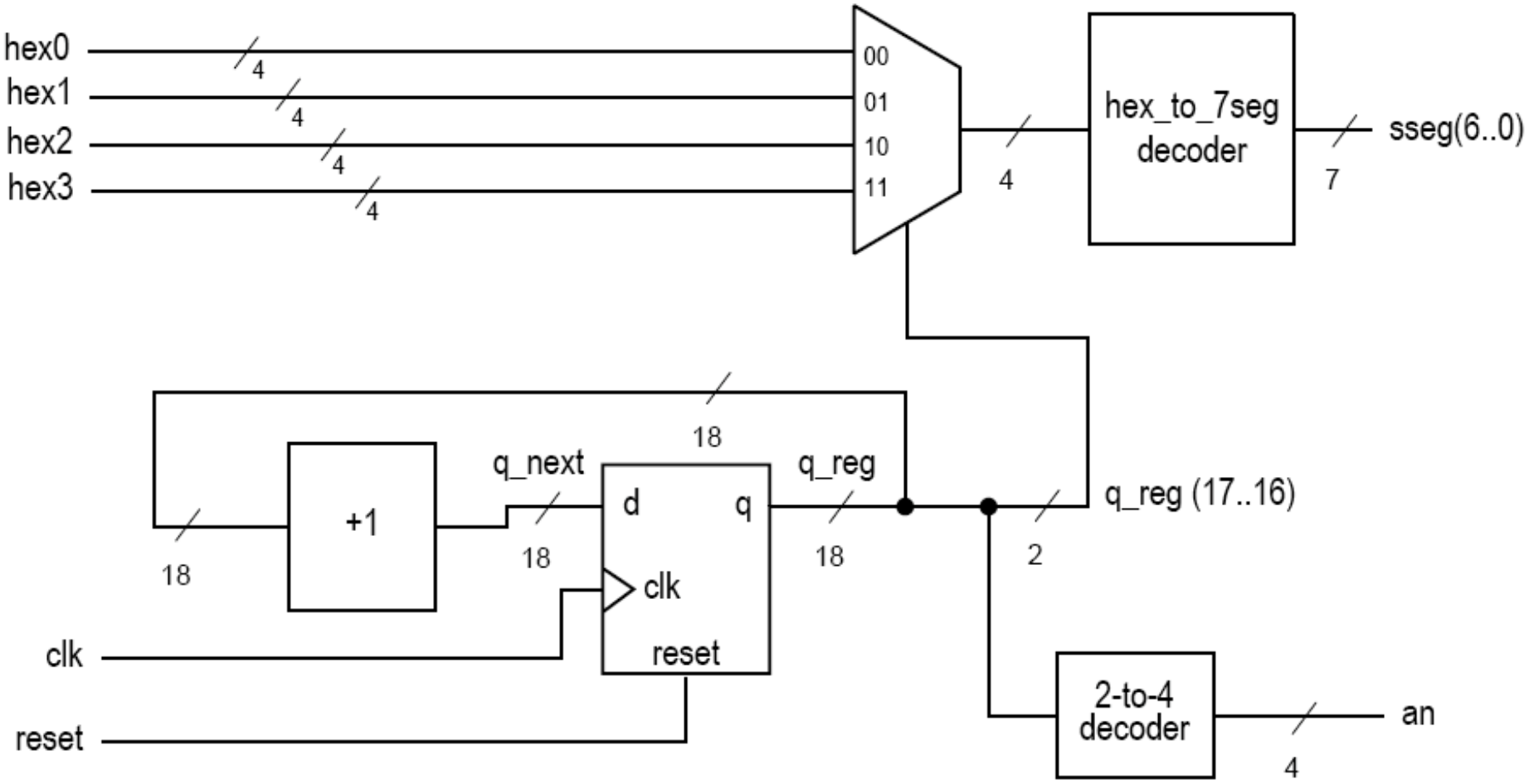
Time-Multiplexed Seven Segment Display



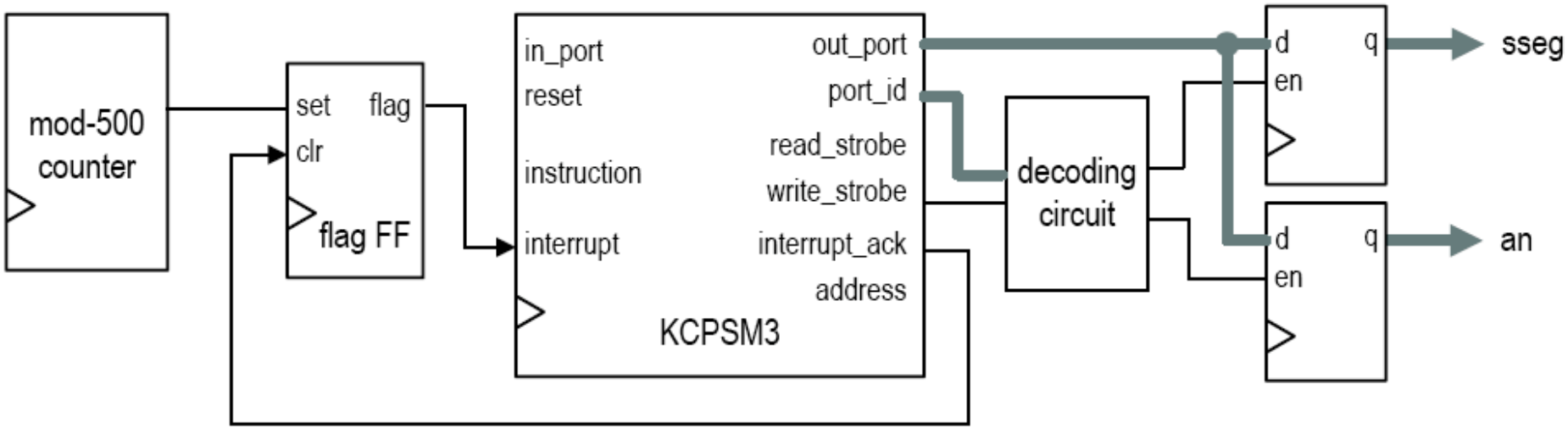
Data inputs



Time-Multiplexing Circuit



Multiplexing Circuit Based on PicoBlaze



- mod-500 generates an interrupt every 5us.
- ISR loads **sseg[6:0]** from memory and generates corresponding port ID for **an[3:0]**.