Introduction to Hardware Design
Memory and Basic Processor Interfaces

Profs. Sundeep Rangan, Siddharth Garg

1. *Sizing memory.* Suppose that a communications receiver receives data at a rate of 1 Gsps (giga sample per second) from each antenna. The receiver has 4 antennas and each sample is a complex number represented by 12 bits for the real part and 12 bits for the imaginary part (i.e., 24 bits per sample).

   (a) What is the total data rate in Gbps (giga bits per second) from all antennas?

   (b) Suppose there is a sample buffer of size 8 MB. How much time (in milliseconds) can the buffer hold before it is full?

   (c) Suppose that each memory can be written at a rate of 400 MHz with a bus width of 64 bits. How many such memories are needed to handle the data rate from part (a)? Assume that you cannot split a sample across multiple memories and you cannot split the real or imaginary component of a sample ver multiple writes.

   Solution:

   Enter your solutions here.

2. *FIR filter*: Consider implementing an FIR filter or convolution. An FIR filter takes an input signal $x[n]$, $n = 0, \ldots, N-1$ and produces an output signal $y[n]$ via

$$y[n] = \sum_{k=0}^{K-1} h[k]x[n-k]$$

   where $h[0], \ldots, h[K-1]$ are the filter coefficients. Suppose all values are 16-bit signed integers. Assume the arrays are length $N = 1000$ and the filter length is $K = 32$.

   (a) What is the number of 32-bit words needed to store x, y and h?

   (b) Suppose that the filter coefficients are stored in registers, and each 32-bit word write takes 5 clock cycles to write. How many clock cycles are needed to program the filter coefficients?

   (c) Suppose that FIR filter IP can run the filter operation for N length vectors in $N + K$ clock cycles. How many clock cycles are needed to run the filter for length $N = 1000$ vectors and $K = 32$ filter coefficients?

   (d) If the processing requires programming the filter coefficients, then running the filter, what percentage of the total time is spent programming the filter coefficients?

Solution:

Enter your solutions here.

3. *AXI4-Lite Write*: Suppose a processor (master) writes to a hardware IP (slave) using the AXI4-Lite protocol. Assume the following AXI4-Lite write timing:

   - On cycle 1, the master asserts `AWVALID` and `WVALID` with address on `AWADDR` and data on `WDATA`.
   - On cycle 2, the slave asserts `AWREADY`.
   - On cycle 3, the slave asserts `WREADY`.
   - There is a 2-cycle delay from the time *both* the address and data have been transferred to when the slave asserts `BVALID`.
   - There is a 1-cycle delay from the time `BVALID` is asserted to when the master asserts `BREADY`.

   Answer the following questions based on the timing above:

   (a) On which cycle is the address transferred on the AW channel?

   (b) On which cycle is the data transferred on the W channel?

   (c) On which cycle is `BVALID` asserted by the slave?

   (d) On which cycle is `BREADY` asserted by the master?

   (e) For each channel separately, on which cycle can the master first de-assert: `AWVALID` and `WVALID`?

   (f) For each channel separately, on which cycle can the master first assert `AWVALID` and `WVALID` for the *next* write transaction? If the master wants to start the next write with both `AWVALID` and `WVALID` high in the same cycle, what is the earliest such cycle?

Solution:

Enter your solutions here.

4. *AXI4-Lite Read*: A processor (master) performs a read from a hardware IP (slave) using the AXI4-Lite protocol. The following timing behavior occurs:

   - Cycle 1: The master asserts `ARVALID` with the read address on `ARADDR`.
   - Cycle 2: The slave asserts `ARREADY`.
   - Cycle 3: The slave asserts `RVALID` with the read data on `RDATA`.
   - Cycle 4: The master asserts `RREADY`.
   - The slave has a 1-cycle latency from the time the address is transferred to the time it asserts `RVALID`.
   - The master has a 1-cycle delay from the time `RVALID` is asserted to the time it asserts `RREADY`.

Answer the following:

(a) On which cycle is the read address transferred?

(b) On which cycle is the read data transferred?

(c) On which cycle may the master de-assert `ARVALID`?

(d) On which cycle may the slave de-assert `RVALID`?

(e) On which cycle may the master begin the next read transaction (i.e., assert `ARVALID` for the next read)?

Solution:

Enter your solutions here.

5. *IP interface*: Suppose an IP implements the iterative update:

$$x[n+1] = f(x[n], a, b)$$

where $f(x, a, b)$ is some known function, and a and b are constants. The IP is given an initial value $x[0]$ and returns the final value $x[N]$ after a specified number of iterations N. Assume all values are 32-bit signed integers.

(a) You decide to write a Vitis HLS function to implement this IP.

```
void diff_eq_solver(...)

#pragma HLS INTERFACE s_axilite port=... bundle=CTRL_BUS
// ...
```

What arguments would you include in the function, and what #pragma HLS INTERFACE lines would you write? You do *not* need to write the body of the function.

(b) Vitis HLS will automatically build the AXI4-Lite interface. Assuming it adds a 32-bit `AP_CTRL` register, what is the register map for this IP? Assume the base address is 0x00 and each register is 32 bits wide.

Solution:

Enter your solutions here.