integrated into the network that interconnects the processing nodes. A system may be configured with up to one I/O channel for every eight PEs. All channels are accessible and controllable from all PEs.

Further information about this "scalable" I/O architecture can be found in Cray Research (1995c).

## EXERCISES

### Section 1.3

1. Write a sequence of instructions for SIC to set ALPHA equal to the product of BETA and GAMMA. Assume that ALPHA, BETA, and GAMMA are defined as in Fig. 1.3(a).

2. Write a sequence of instructions for SIC/XE to set ALPHA equal to 4 * BETA – 9. Assume that ALPHA and BETA are defined as in Fig. 1.3(b). Use immediate addressing for the constants.

3. Write a sequence of instructions for SIC to set ALPHA equal to the integer portion of BETA ÷ GAMMA. Assume that ALPHA and BETA are defined as in Fig. 1.3(a).

4. Write a sequence of instructions for SIC/XE to divide BETA by GAMMA, setting ALPHA to the integer portion of the quotient and DELTA to the remainder. Use register-to-register instructions to make the calculation as efficient as possible.

5. Write a sequence of instructions for SIC/XE to divide BETA by GAMMA, setting ALPHA to the value of the quotient, rounded to the nearest integer. Use register-to-register instructions to make the calculation as efficient as possible.

6. Write a sequence of instructions for SIC to clear a 20-byte string to all blanks.

7. Write a sequence of instructions for SIC/XE to clear a 20-byte string to all blanks. Use immediate addressing and register-to-register instructions to make the process as efficient as possible.

8. Suppose that ALPHA is an array of 100 words, as defined in Fig. 1.5(a). Write a sequence of instructions for SIC to set all 100 elements of the array to 0.

9. Suppose that ALPHA is an array of 100 words, as defined in Fig. 1.5(b). Write a sequence of instructions for SIC/XE to set all 100

elements of the array to 0. Use immediate addressing and register-to-register instructions to make the process as efficient as possible.

10. Suppose that RECORD contains a 100-byte record, as in Fig. 1.7(a). Write a subroutine for SIC that will write this record onto device 05.

11. Suppose that RECORD contains a 100-byte record, as in Fig. 1.7(b). Write a subroutine for SIC/XE that will write this record onto device 05. Use immediate addressing and register-to-register instructions to make the subroutine as efficient as possible.

12. Write a subroutine for SIC that will read a record into a buffer, as in Fig. 1.7(a). The record may be any length from 1 to 100 bytes. The end of the record is marked with a "null" character (ASCII code 00). The subroutine should place the length of the record read into a variable named LENGTH.

13. Write a subroutine for SIC/XE that will read a record into a buffer, as in Fig. 1.7(b). The record may be any length from 1 to 100 bytes. The end of the record is marked with a "null" character (ASCII code 00). The subroutine should place the length of the record read into a variable named LENGTH. Use immediate addressing and register-to-register instructions to make the subroutine as efficient as possible.