Computer Architecture

A Generic CPU - 3

Dr. Masood Mir

Week 6

A Generic Datapath

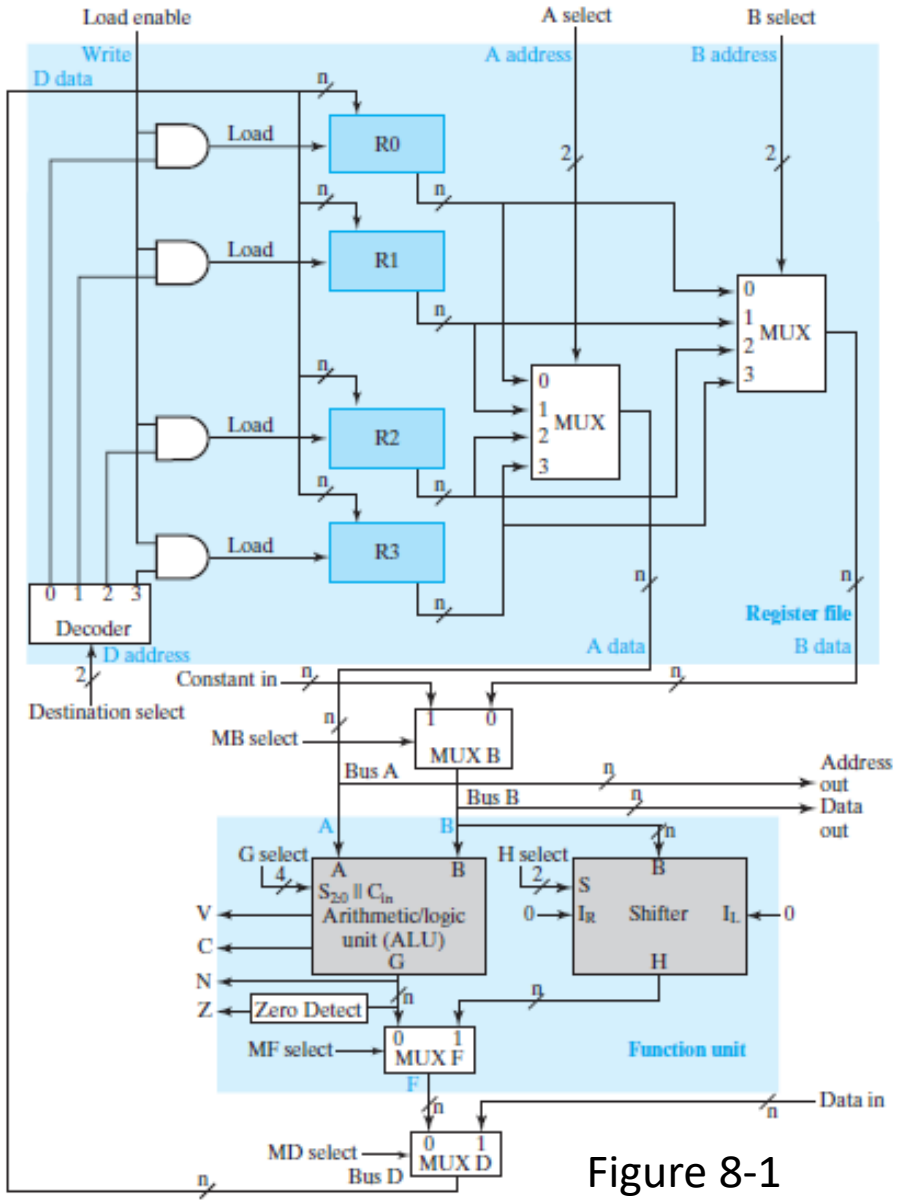

Figure 8-1

- The original datapath we discussed can be reduced. This reduction is important, since we frequently use this datapath.
- We also need to have hierarchy in the design, so that we can replace one module with another, so that we are not tied to specific logic implementations.
- A typical datapath has more than four registers. CPUs with 32 or more registers are common.
- The construction of a bus system with a large number of registers requires different techniques. A set of registers having common microoperations performed on them may be organized into a register file.
- The typical register file is a special type of fast memory that permits one or more words to be read and one or more words to be written, all simultaneously.

Making Datapath more generic

- Functionally, a simple register file contains the equivalent of the logic shaded in blue in Figure 8-1. Due to the memory-like nature of register files, the A select, B select, and Destination select inputs in the figure become three addresses.
- A address accesses a word to be read onto A data, the B address accesses a second word to be read onto B data, and the D address accesses a word to be written into from D data. All of these accesses occur in the same clock cycle.
- A Write input corresponding to the Load Enable signal is also provided. When at 1, the Write signal permits registers to be loaded during the current clock cycle, and when at 0, prevents register loading.
- The size of the register file is 2^m * n where m is the number of register address bits and n is the number of bits per register.

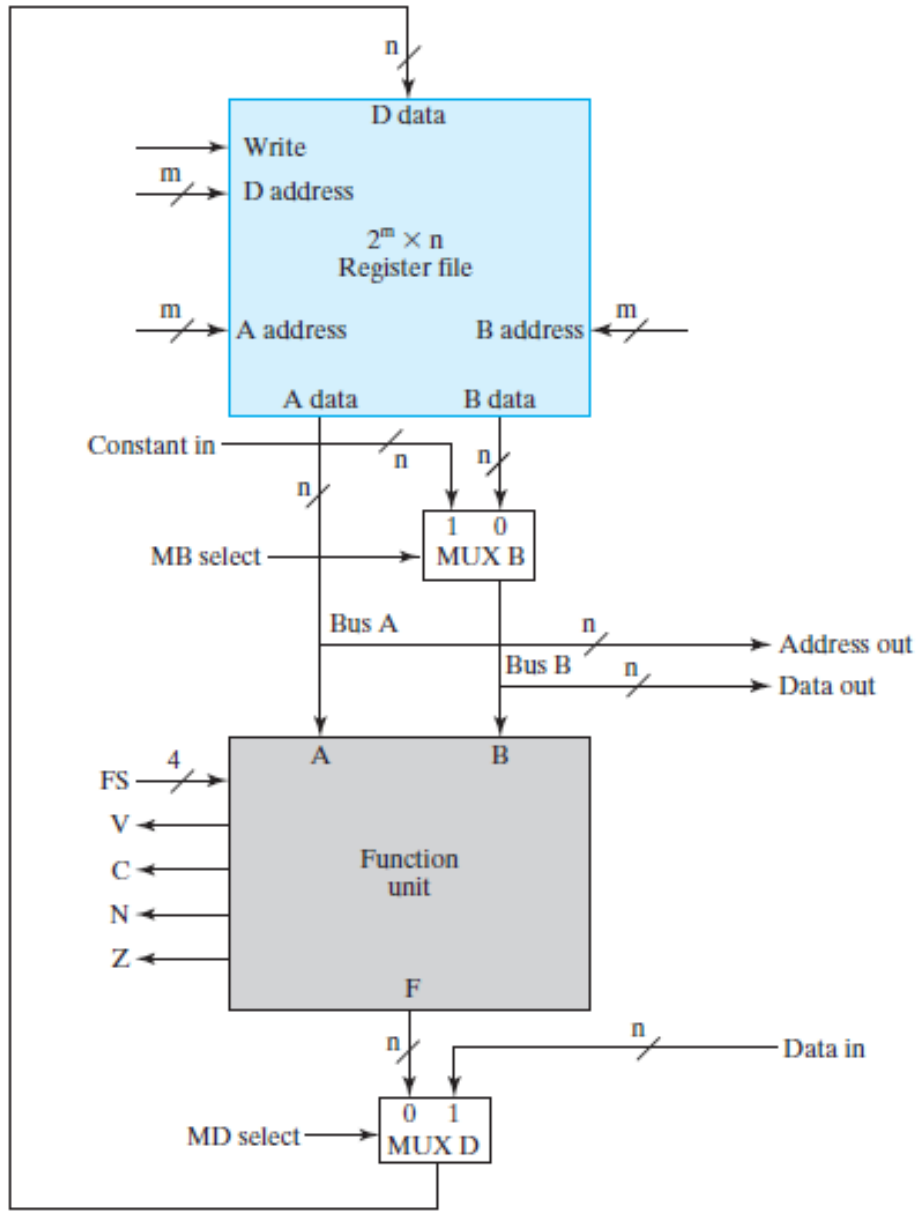

FIGURE 8-10
Block Diagram of Datapath Using the Register File and Function Unit

Assignment - 1

- Make a detailed diagram of figure 8-10 as figure 8-1 with eight registers.

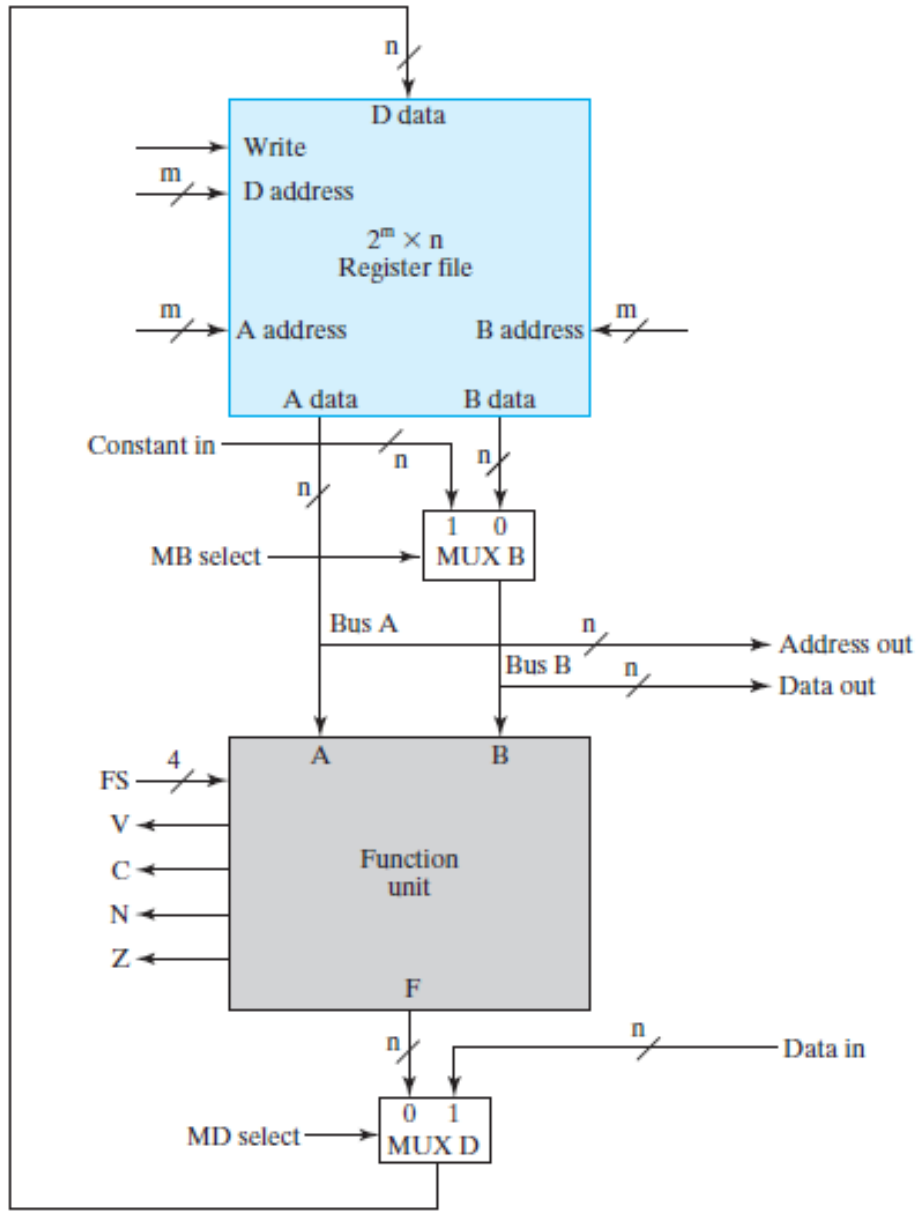

FIGURE 8-10
Block Diagram of Datapath Using the Register File and Function Unit

Making Datapath more generic

- Since the ALU and the shifter are shared processing units with outputs that are selected by MUX F, it is convenient to group the two units and the MUX together to form a shared function unit. Gray shading in Figure 8-1 highlights the function unit, which can be represented by the symbol given in Figure 8-10.
- The inputs to the function unit are from Bus A and Bus B, and the output of the function unit goes to MUX D.
- The function unit also has the four status bits V, C, N, and Z as added outputs.
- In Figure 8-1, there are three sets of select inputs: the G select, H select, and MF select. In Figure 8-10, there is a single set of select inputs labeled FS, for "function select."

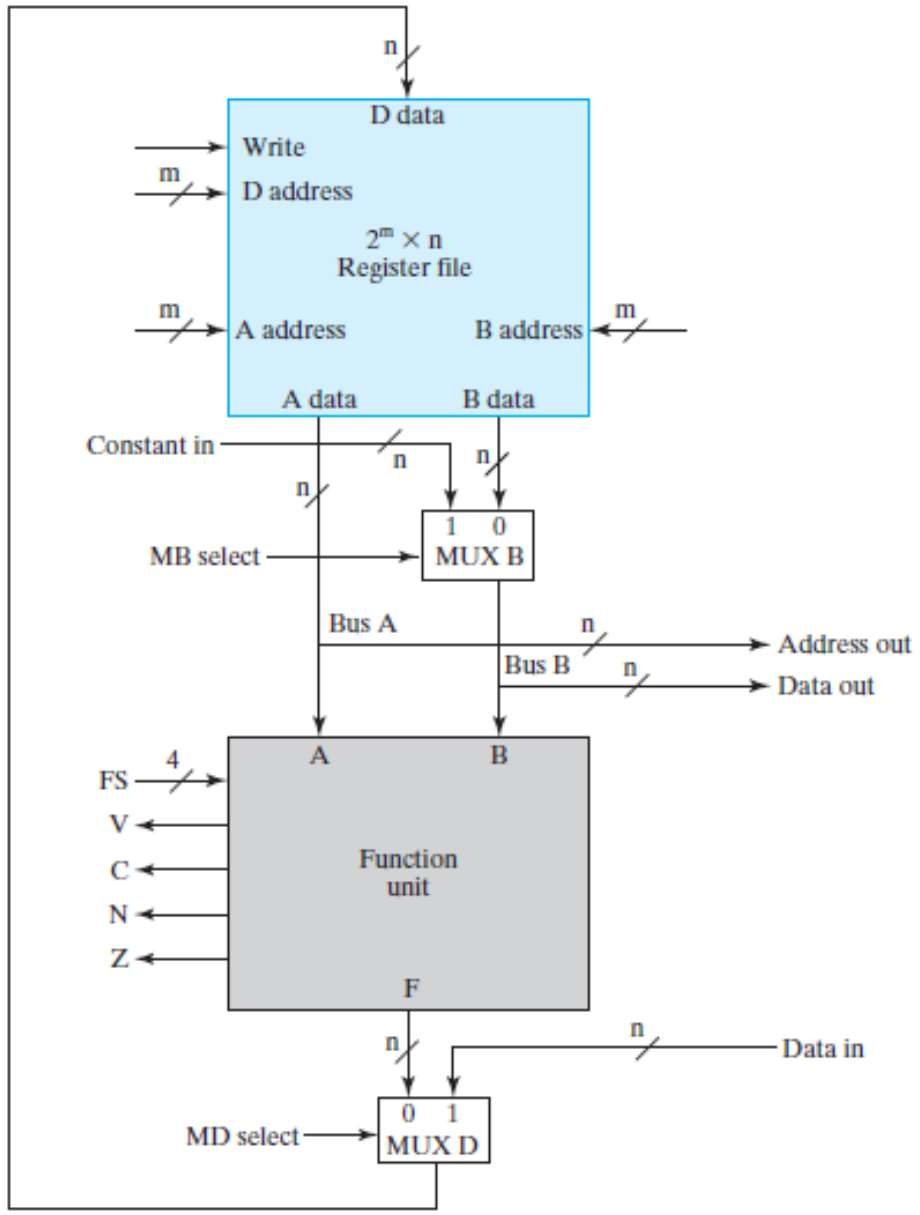

FIGURE 8-10
Block Diagram of Datapath Using the Register File and Function Unit

Making Datapath more generic

- The codes for MF select, G select, and H select must be defined in terms of the codes for FS, as below:

TABLE 8-4
***G* Select, *H* Select, and *MF* Select Codes Defined in Terms of *FS* Codes**

FS(3:0)	MF Select	G Select(3:0)	H Select(3:0)	Microoperation
0000	0	0000	XX	$F = A$
0001	0	0001	XX	$F = A + 1$
0010	0	0010	XX	$F = A + B$
0011	0	0011	XX	$F = A + B + 1$
0100	0	0100	XX	$F = A + \overline{B}$
0101	0	0101	XX	$F = A + \overline{B} + 1$
0110	0	0110	XX	$F = A - 1$
0111	0	0111	XX	$F = A$
1000	0	1X00	XX	$F = A \wedge B$
1001	0	1X01	XX	$F = A \vee B$
1010	0	1X10	XX	$F = A \oplus B$
1011	0	1X11	XX	$F = \overline{A}$
1100	1	XXXX	00	$F = B$
1101	1	XXXX	01	$F = \text{sr } B$
1110	1	XXXX	10	$F = \text{sl } B$

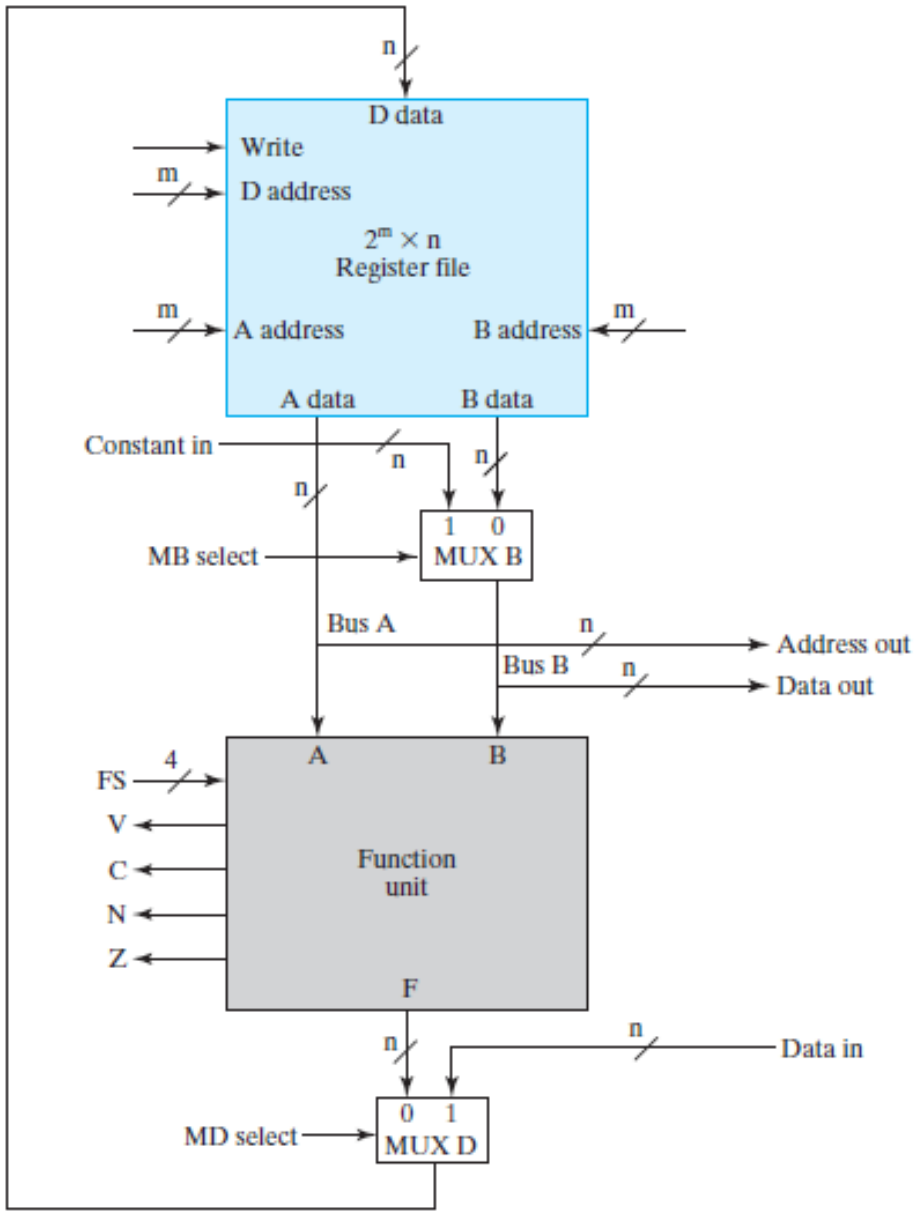

FIGURE 8-10
Block Diagram of Datapath Using the Register File and Function Unit

Making Datapath more generic

- MF is 1 for the leftmost two bits of FS both equal to 1. If MF select = 0, then the G select codes determine the function on the output of the function unit and H select is in don't care. If MF select = 1, then G select will not play any role and the H select codes determine the function on the output of the function unit.
- From Table 8-4, the code transformations can be implemented using the Boolean equations:
- MF = F3.F2, G3 = F3, G2 = F2, G1 = F1, G0 = F0, H1 = F1, and H0 = F0.
- The status bits will not have useful information when the shifter is selected. A more complex system is required to provide status bits for shifter. In present circuit, the status bit implementation is for arithmetic circuit.

TABLE 8-4
***G* Select, *H* Select, and *MF* Select Codes Defined in Terms of *FS* Codes**

FS(3:0)	MF Select	G Select(3:0)	H Select(3:0)	Microoperation
0000	0	0000	XX	$F = A$
0001	0	0001	XX	$F = A + 1$
0010	0	0010	XX	$F = A + B$
0011	0	0011	XX	$F = A + B + 1$
0100	0	0100	XX	$F = A + \overline{B}$
0101	0	0101	XX	$F = A + \overline{B} + 1$
0110	0	0110	XX	$F = A - 1$
0111	0	0111	XX	$F = A$
1000	0	1X00	XX	$F = A \wedge B$
1001	0	1X01	XX	$F = A \vee B$
1010	0	1X10	XX	$F = A \oplus B$
1011	0	1X11	XX	$F = \overline{A}$
1100	1	XXXX	00	$F = B$
1101	1	XXXX	01	$F = \text{sr } B$
1110	1	XXXX	10	$F = \text{sl } B$

Making Datapath more generic

- MF is 1 for the leftmost two bits of FS both equal to 1. If MF select = 0, then the G select codes determine the function on the output of the function unit and H select is in don't care. If MF select = 1, then G select will not play any role and the H select codes determine the function on the output of the function unit.
- From Table 8-4, the code transformations can be implemented using the Boolean equations:
- MF = F3.F2, G3 = F3, G2 = F2, G1 = F1, G0 = F0, H1 = F1, and H0 = F0.
- The status bits will not have useful information when the shifter is selected. A more complex system is required to provide status bits for shifter. In present circuit, the status bit implementation is for arithmetic circuit.

TABLE 8-4
***G* Select, *H* Select, and *MF* Select Codes Defined in Terms of *FS* Codes**

FS(3:0)	MF Select	G Select(3:0)	H Select(3:0)	Microoperation
0000	0	0000	XX	$F = A$
0001	0	0001	XX	$F = A + 1$
0010	0	0010	XX	$F = A + B$
0011	0	0011	XX	$F = A + B + 1$
0100	0	0100	XX	$F = A + \overline{B}$
0101	0	0101	XX	$F = A + \overline{B} + 1$
0110	0	0110	XX	$F = A - 1$
0111	0	0111	XX	$F = A$
1000	0	1X00	XX	$F = A \wedge B$
1001	0	1X01	XX	$F = A \vee B$
1010	0	1X10	XX	$F = A \oplus B$
1011	0	1X11	XX	$F = \overline{A}$
1100	1	XXXX	00	$F = B$
1101	1	XXXX	01	$F = \text{sr } B$
1110	1	XXXX	10	$F = \text{sl } B$

Assignment - 2

- Design a decoder that takes FS(3:0) as input and produces MF Select, G Select(3:0) and H Select(3:0) as output.

TABLE 8-4
***G* Select, *H* Select, and *MF* Select Codes Defined in Terms of *FS* Codes**

FS(3:0)	MF Select	G Select(3:0)	H Select(3:0)	Microoperation
0000	0	0000	XX	$F = A$
0001	0	0001	XX	$F = A + 1$
0010	0	0010	XX	$F = A + B$
0011	0	0011	XX	$F = A + B + 1$
0100	0	0100	XX	$F = A + \overline{B}$
0101	0	0101	XX	$F = A + \overline{B} + 1$
0110	0	0110	XX	$F = A - 1$
0111	0	0111	XX	$F = A$
1000	0	1X00	XX	$F = A \wedge B$
1001	0	1X01	XX	$F = A \vee B$
1010	0	1X10	XX	$F = A \oplus B$
1011	0	1X11	XX	$F = \overline{A}$
1100	1	XXXX	00	$F = B$
1101	1	XXXX	01	$F = \text{sr } B$
1110	1	XXXX	10	$F = \text{sl } B$

Extra Slide

- .