Embedded Real-Time Systems (AME 3623)
Homework 3 Solutions

May 2, 2005

Question 1

1. Describe (in brief) the function of the status register.

   The status register captures (in part) the state of the processor. In particular, it captures information related to the result of the most recently executed instruction (or instructions). For example, the **zero bit** *is set if the most recent arithmetic operation resulted in a zero value, and the* **carry bit** *indicates whether the last arithmetic operation resulted in a carry.*

2. Describe the similarities and differences between the Atmel Mega8 RAM and its general purpose registers.

   Similarities: the registers and the RAM store values in memories that are 8 bits wide; because there are multiple memory elements in each, the processor must generate an address to select the appropriate element.

   Differences: there are fewer general-purpose registers than memory elements in the RAM (specifically, there are 32 bytes vs. 1024 bytes); this means that there are fewer address lines for the registers (5 vs. 10). In addition, access to the registers is faster than the RAM: we can perform a register read and write within a single clock cycle, but can perform only one of RAM read/write during the same period of time.

Question 2

1. Name the four distinct components that **can** compose a serial data frame. Describe the function of each.

   (a) *Start bit: indicates the beginning of the frame. The beginning of the start bit is used by the receiver to synchronize its clock with the sender's clock (so that it is clear to the receiver when one bit of the frame ends and the next begins).*

   (b) *Data bits: contain the data to be transferred.*

   (c) *Parity bit: indicates whether there is an odd or even number of 0 bits (or 1 bits) within the set of data bits. This bit is used as a first level of error checking: both the sender and receiver independently compute this value given the data bits; if the two values do not match, then the frame is considered to be invalid.*

   (d) *Stop bit(s): indicates the end of the frame. This is also used as a mechanism for error checking: if the known stop bit value is not found, then we assume that the frame is invalid.*

2. Name the three levels of serial processing. Describe in brief the approach to preventing and/or detecting errors at each of these levels.

   (a) **Bit level:** *The value of a bit is "broadcast" over a period of time. This allows the receiver to take multiple, independent samples. This (hopefully) improves the probability of interpreting the bit value correctly.*

   (b) **Frame level:** *At the frame level, we use a* **parity bit** *to check the validity of the set of data bits, and we use the* **stop bits** *to confirm that we have properly identified the beginning and end of the frame.*

   (c) **Packet level:** *We use constant values within the packet (typically at the beginning and the end) so that the receiver knows when a new packet has started and ended. In addition, we can use a* **checksum byte** *to confirm the validity of the remaining bytes within the packet (as with the parity bit, the sender and receiver independently compute a checksum value which are then compared by the receiver).*

Question 3

1. Given a ROM module made up of 2048 bytes. How many input lines are necessary? List them.

   Assumption: this is an 8-bit wide memory.

   (a) *2048 bytes require 11 address lines.*

   (b) *8 lines for the data.*

   (c) *1 line for chip select.*

   So: a total of 20 lines are required for the chip.

   (Note that we do not need a read/write line since the processor can only read from this "read only memory")

Question 4

The OUmel mega8 processor has a similar architecture as the Atmel mega8. It supports a set of instructions that are encoded as follows:

Assembly	Description	Machine code
AND	bit-wise AND	0 0 1 0 0 0 r d d d d d r r r r
ADD	add with carry	0 0 0 0 1 1 r d d d d d r r r r
INC	increment	1 0 0 1 0 1 0 d d d d d 0 0 1 1
SWAP	swap nybbles	1 0 0 1 0 1 0 d d d d d 0 0 1 0

Assume the bits are arranged with the most significant bit (I15) on the left.

Assume an arithmetic logical unit with two control bits ($C1$ and $C0$). Assume the following bit combinations for the following operations:

C1	C0	Operation
0	0	AND
0	1	ADD
1	0	INC
1	1	SWAP

1. Give the algebraic expression for the circuit that uniquely identifies each of the four instructions.

   Recall that the "r" and "d" bits specify which registers are involved in the operation; these bits can take on arbitrary values. Hence, we need to only recognize the specific configuration of 0s and 1s in the above table.

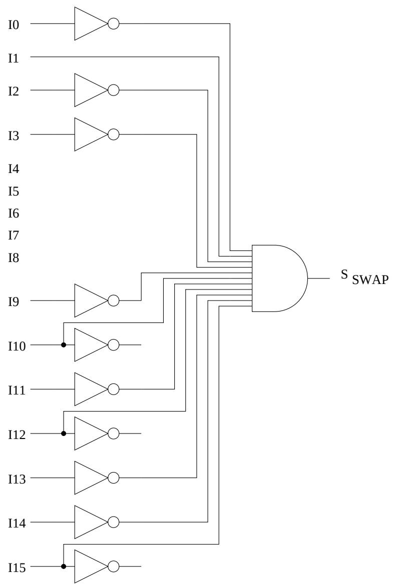
$$
\begin{aligned}
S_{AND} &= \overline{I_{15}} * \overline{I_{14}} * I_{13} * \overline{I_{12}} * \overline{I_{11}} * \overline{I_{10}} \\
S_{ADD} &= \overline{I_{15}} * \overline{I_{14}} * \overline{I_{13}} * \overline{I_{12}} * I_{11} * I_{10} \\
S_{INC} &= I_{15} * \overline{I_{14}} * \overline{I_{13}} * I_{12} * \overline{I_{11}} * I_{10} * \overline{I_9} * \overline{I_3} * \overline{I_2} * I_1 * I_0 \\
S_{SWAP} &= I_{15} * \overline{I_{14}} * \overline{I_{13}} * I_{12} * \overline{I_{11}} * I_{10} * \overline{I_9} * \overline{I_3} * \overline{I_2} * I_1 * \overline{I_0}
\end{aligned}
$$

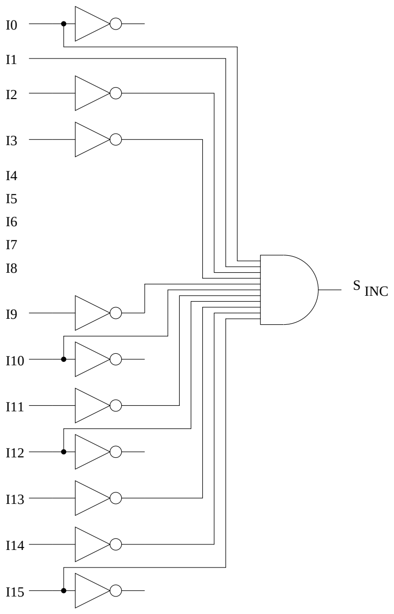
2. Show the circuit that takes as input $I15...I0$ and produces the bits $C1$ and $C0$ for input into the ALU.

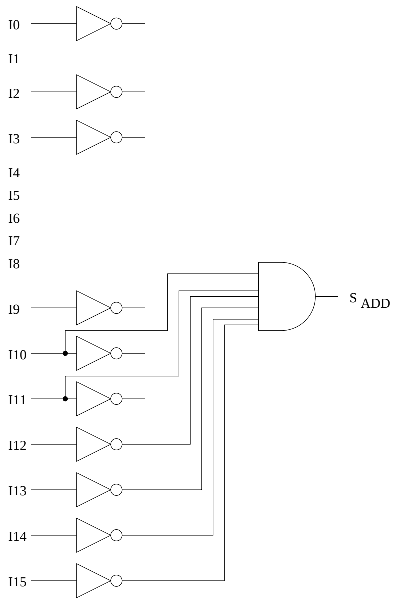
   $C1$ is high for both the INC and SWAP operators. $C0$ is high for ADD and SWAP. So:

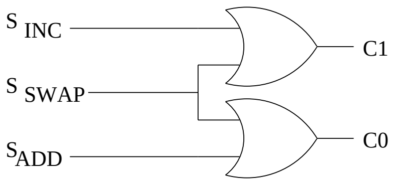
$$
\begin{aligned}
C0 &= S_{ADD} + S_{SWAP} \\
C1 &= S_{INC} + S_{SWAP}
\end{aligned}
$$

   The complete circuit is as follows (the different components have been split apart for simplicity):


I0
I1
I2
I3
I4
I5
I6
I7
I8
I9
I10
I11
I12
I13
I14
I15
S SWAP

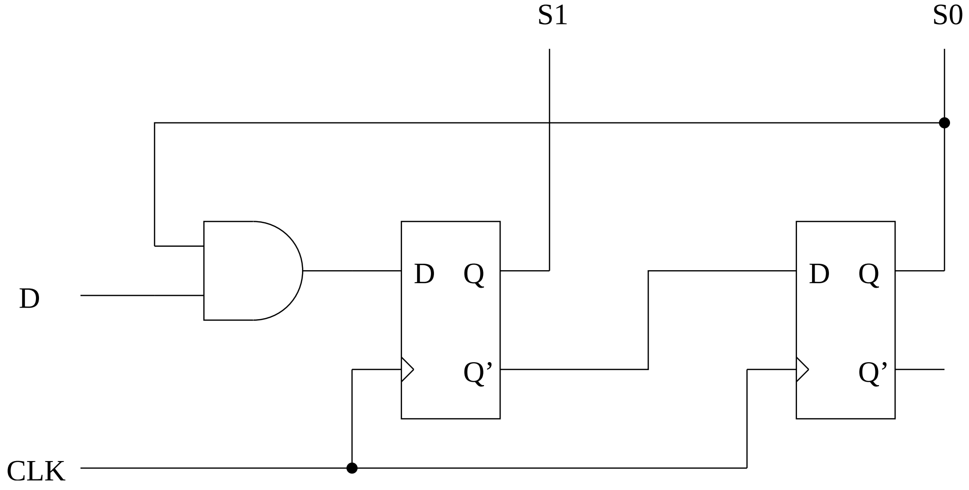

I0
I1
I2
I3
I4
I5
I6
I7
I8
I9
I10
I11
I12
I13
I14
I15
S INC


I0
I1
I2
I3
I4
I5
I6
I7
I8
I9
I10
I11
I12
I13
I14
I15
S ADD

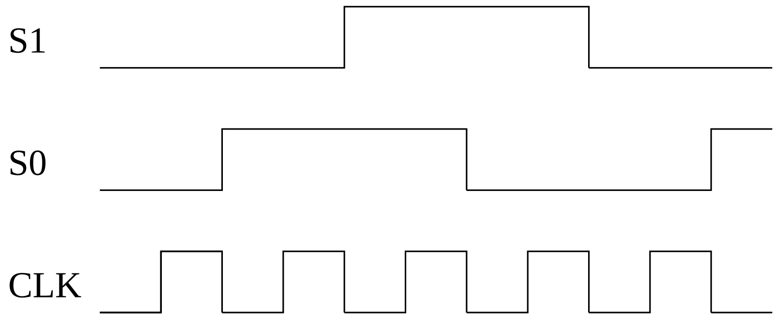

S INC
S SWAP
SADD
C1
C0

Question 5

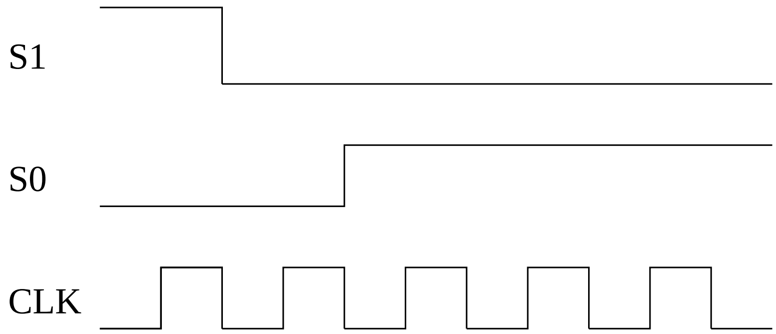
Consider the following circuit:



1. Assume $D = 1$ and an initial condition of $S1 = 0$ and $S0 = 0$. Show the timing diagram for 5 complete clock cycles.



2. Assume $D = 0$ and an initial condition of $S1 = 1$ and $S0 = 0$. Show the timing diagram for 5 complete clock cycles.



Note that in this case, the system does not visit all possible states from this initial condition.

Consider the finite state machine representation of this circuit.

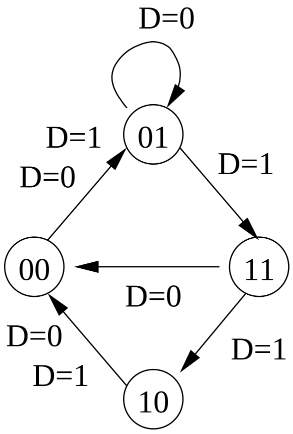
3. What are the states?

   The states are all possible combinations of the two flip flops: 00, 01, 10, 11.

4. What are the events?

   The events correspond to $D = 0$ and $D = 1$, and arrive at the downward edge of the clock.

5. Show the corresponding state transition diagram.



Question 6

1. What two conditions must be met for us to have a "shared data problem?"

   (a) *The execution of one segment of code can be interrupted by the execution of another segment of code.*

   (b) *The two segments of code make use of the same variables (with at least one of the segments writing to at least one of the variables).*

2. What are the advantages and disadvantages of polling over interrupts.

   Interrupts over polling:

   (a) *The main body of code does not have to deal with the timing requirements involved in handling an event.*

   (b) *The main body of code allows us to perform other tasks while we are waiting for an event.*

   (c) *An interrupt handler can address an interrupt almost as soon as it occurs (with some delay due to decoding the interrupt and starting the execution of the handler)).*

   Polling over interrupts:

   (a) *No shared data problem to contend with.*

   (b) *Polling *can* respond quicker than interrupts if we are exclusively busy waiting (executing an interrupt requires several instructions before the handler code is executed).*