REVIEW QUESTIONS

Spend time reviewing your lecture notes from the previous week. **Prepare** written answers to these questions at the top of a Microsoft Word document.

1. **State** how flip flops and latches differ in terms of where in the clock period each memory type samples the data input.
2. **Describe** how Moore Machines and Mealy Machines differ in terms of equation parameters.
3. **State** what the arrow departing a state in a state diagram implies. In other words, these arcs represent something critically important to sequential machines. What is that important idea?
4. **Calculate** the number of arrows that would depart each state of a machine that has three inputs called A, B, and C.
5. **State** the two ways that we have seen to represent the **reset event** in a state diagram. In other words, what are the two ways that reset is drawn in state diagrams?

TODAY'S PROBLEM STATEMENT

Thanks to Dr. Livingston for suggesting beat generators as neat FSMs to explore.

A modern pop music band uses acoustic and electronic instruments including guitars, keyboards, and drum sets. The acoustic instruments are played by the musicians and create sound with air, strings, taut drum heads, and cymbals. The electronic instruments include electric guitars played by musicians, of course, but also include electronic beat generators. Beat generators are machines that create electronic sounds that are mixed in with the acoustic sounds played by the musicians.

Pop, rock, country, and blues music styles most commonly create rhythm as four music notes (or beats) per music bar. Don't worry if you don't read music. Just know that four beats are the basic "set of notes" for these styles of music. These "sets of notes" then form rhythmic and melodic phrases when strung together. Consider a drum beat as the word "boom" and a cymbal metallic ring as the word "chick." Some common rhythms in pop and rock music are:

- Boom-chick-boom-chick,
- Boom-chick-chick-chick, and
- Boom-chick-chick-boom

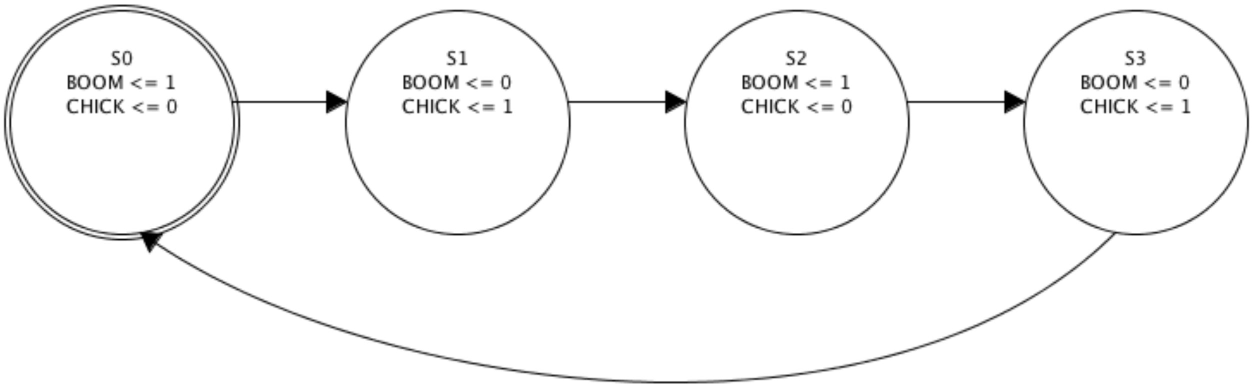
Consider a Moore Machine that is used to generate the first sequence of beat sounds. The band sets the speed of the rhythm by varying the machine clock frequency.

REVIEW THE PAPER AND PENCIL DESIGN AND IMPLEMENTATION STEPS

1. **Create** state diagram.
2. **Assign** binary numbers to each state name. This will be the **stored machine state**.
3. **Create** truth tables for combinational logic blocks in the machine model: next state, output
4. **Minimize** the next state and output equations using K-maps or Boolean Algebra techniques.
5. **Draw** the circuit schematic on paper or in a CAD tool.
6. **Verify** the solution using either paper simulation or CAD tool simulation.
7. **Build** the physical circuit.
8. **Test** the physical circuit to ensure correct construction.

STEP 1: CREATE STATE DIAGRAM

Let's design and build a simple Moore Machine that generates the first rhythm. **Assume** that the beat generator produces binary outputs BOOM and CHICK that control a sound synthesizer. Assume that BOOM=1 means that the sound synthesizer should produce the drum sound and CHICK=1 means that the sound synthesizer should produce the cymbal sound. Because there are four beats per musical bar, let's assign one state per beat.



STEP 2: ASSIGN BINARY NUMBERS TO EACH STATE NAME

Four states require two state memory bits because $\log_2 4 = 2$. Thus four binary numbers — one for each state — will be stored through time as the machine progresses through its sequence. Let's assign binary number 00 to S0, binary number 01 to S1, binary number 10 to S2, and binary number 11 to S3. This is called **standard binary encoding** because the state names suggest the encoded stored value. **Remember** that stored values are the Q outputs of flip flops.

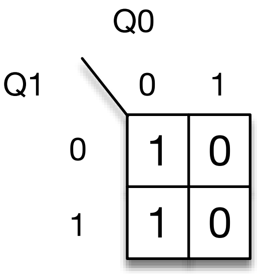
STATE NAME	Q1	Q0
S0	0	0
S1	0	1
S2	1	0
S3	1	1

STEPS 3 AND 4: CREATE TRUTH TABLES AND EQUATIONS

Let’s do the output combinational logic block first. **Remember** that the output logic depends only on the current stored state: BOOM(Q) and CHICK(Q). We can place both outputs in the same truth table.

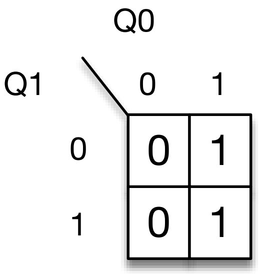
	INPUTS		**OUTPUTS**	
STATE NAME	**Q1**	**Q0**	**BOOM**	**CHICK**
S0	0	0	1	0
S1	0	1	0	1
S2	1	0	1	0
S3	1	1	0	1

The solution to the equations can be found using Boolean algebra of the K-map technique.



Q1 \ Q0	0	1
0	1	0
1	1	0

BOOM = Q0_



Q1 \ Q0	0	1
0	0	1
1	0	1

CHICK = Q0

Now repeat the process for the other combinational logic block — the block creating the next state equations D1 and D0 — one data input for each memory flip flop. This block is always a little more complicated because it requires any additional machine inputs and it also requires us to keep **current state stored in the memory (Q)** separate in our minds from **next state to be transitioned to and stored (D).**

Remember that the flip flop data inputs are functions of Q and I: D(Q,I).
In this machine there no other inputs so D only depends on Q: D(Q).

	INPUTS		**OUTPUTS**	
STATE TRANSITION	**Q1**	**Q0**	**D1**	**D0**
S0 —> S1	0	0	0	1
S1 —> S2	0	1	1	0
S2 —> S3	1	0	1	1
S3 —> S0	1	1	0	0

The truth table shows the **current state feedback signal inputs** Q1 and Q0.
The truth table shows the **next state outputs** D1 and D0.
The standard binary encodings are complete for current and next state values.
The K-maps for D1 and D0 and minimized equations are:

Q1 \ Q0	0	1
0	0	1
1	1	0

D1 = Q1 xor Q0

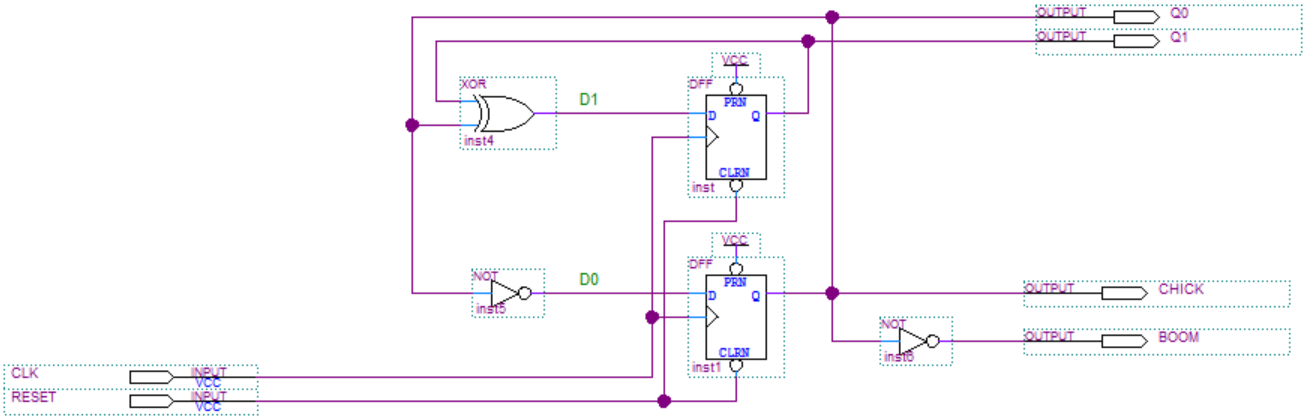
Q1 \ Q0	0	1
0	1	0
1	1	0

D0 = Q0_

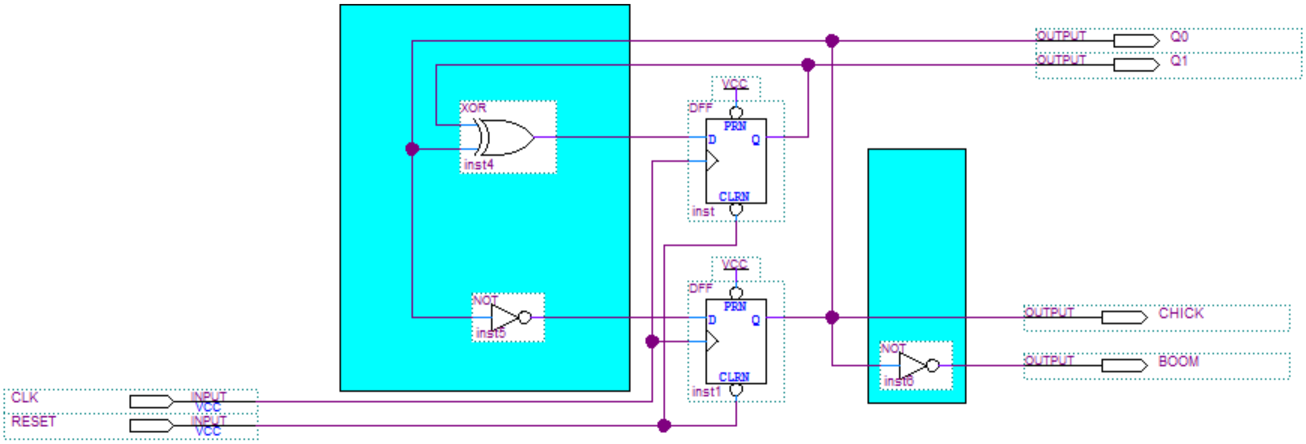
STEP 5: DRAW THE CIRCUIT ON PAPER OR IN A CAD TOOL

Let's use Quartus as the CAD tool for schematic and simulation. The Quartus schematic is shown below. **Note** that the D flip flop element has both preset and reset signals. Preset is rarely used and most often the preset signals are simply tied to the non-active level to ensure that preset never occurs. In this case, bubble symbology shows preset active with logic-0 so the presets are tied to the non-active voltage VCC. Standard practice also has the flip flops organized with Q1 at the top of the diagram followed by Q0 below it.

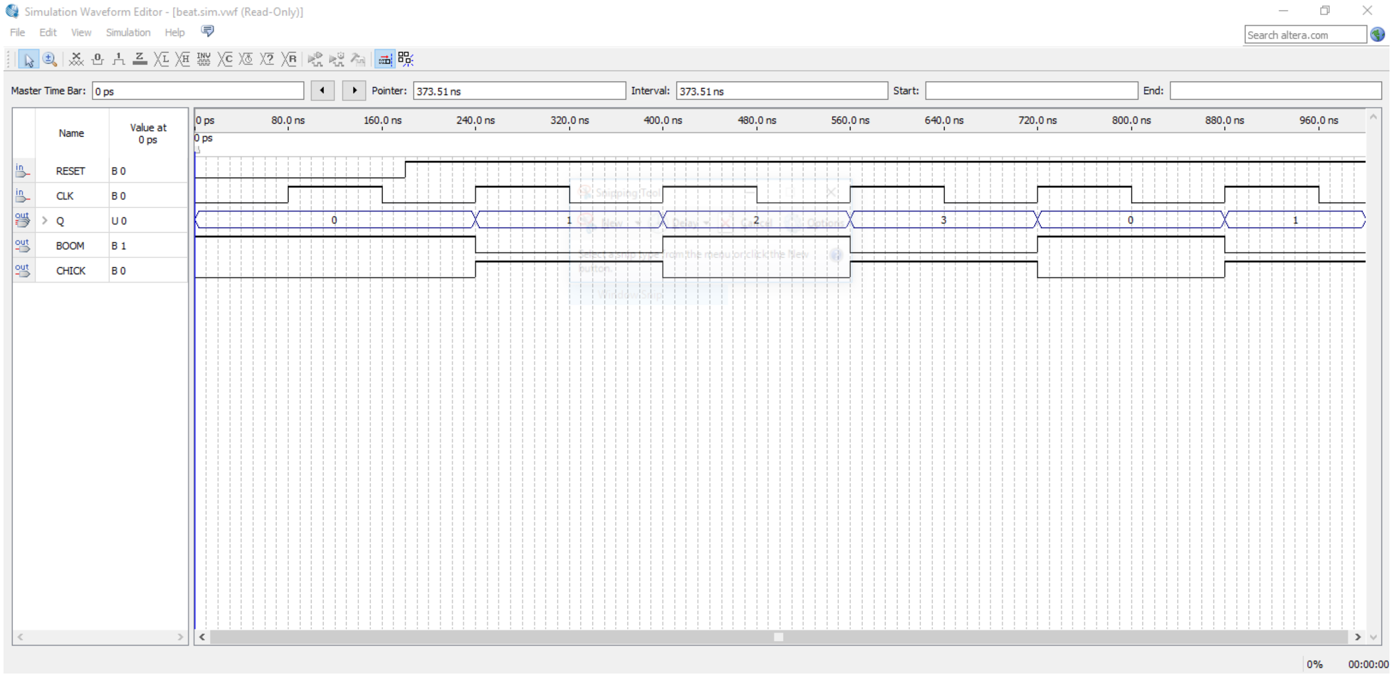
The Moore Machine Model is nicely implemented. The next state combinational logic is in place on the left hand side of the diagram. The state memory is in the middle. The output combinational logic is on the right hand side of the diagram. The common clock and reset signals are attached.



Here is the circuit again with the combinational logic blocks highlighted to make them more visible.



Simulation in Quartus follows standard practice. The inputs RST and CLK are added along with the outputs from each combinational logic block: Q1, Q0, BOOM, and CHICK. I usually run simulation for two to three complete machine cycles so that the periodic sequential behavior can be visualized and verified — in this case, a BOOM occurs in S0 and S2 while a CHICK occurs in S1 and S3.


Simulation Waveform Editor - [beat.sim.vwf (Read-Only)]

Master Time Bar: 0 ps

Pointer: 373.51 ns

Interval: 373.51 ns

Start:

End:

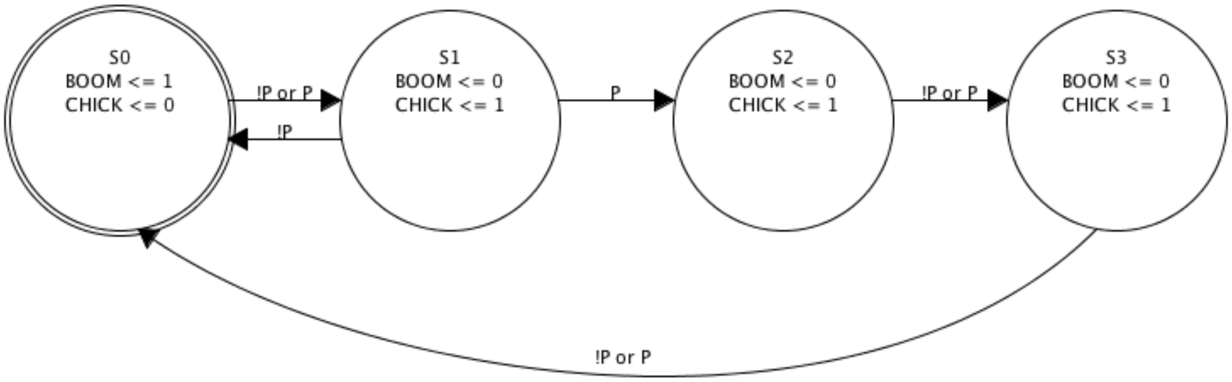
Name	Value at 0 ps
RESET	B 0
CLK	B 0
Q	U 0
BOOM	B 1
CHICK	B 0

Now, let's modify the machine so that two different rhythm patterns can be generated. Let's use an input called P to direct arrows along the machine to generate the first and second patterns:

- Boom-chick-boom-chick
- Boom-chick-chick-chick

The band controls the selected rhythm with a foot pedal mounted on stage.

STEP 1: CREATE STATE DIAGRAM



The new state diagram shows the new control input on each arrow. The control input determines which arrow gets followed. The machine takes advantage of the fact that the boom-chick-boom-chick rhythm is really a cycle back and forth between S0 and S1. But, if the rhythm machine enters boom-chick-chick-chick it cannot be stopped until the group of beats is finished and the machine has returned to S0 to check for a different user request.

STEP 2: ASSIGN BINARY NUMBERS TO EACH STATE NAME

Let’s use standard binary encoding again.

STATE NAME	Q1	Q0
S0	0	0
S1	0	1
S2	1	0
S3	1	1

STEPS 3 AND 4: CREATE TRUTH TABLES AND EQUATIONS

The output only depends on current state. This example has different outputs from the first machine and thus we need to redesign outputs BOOM and CHICK.

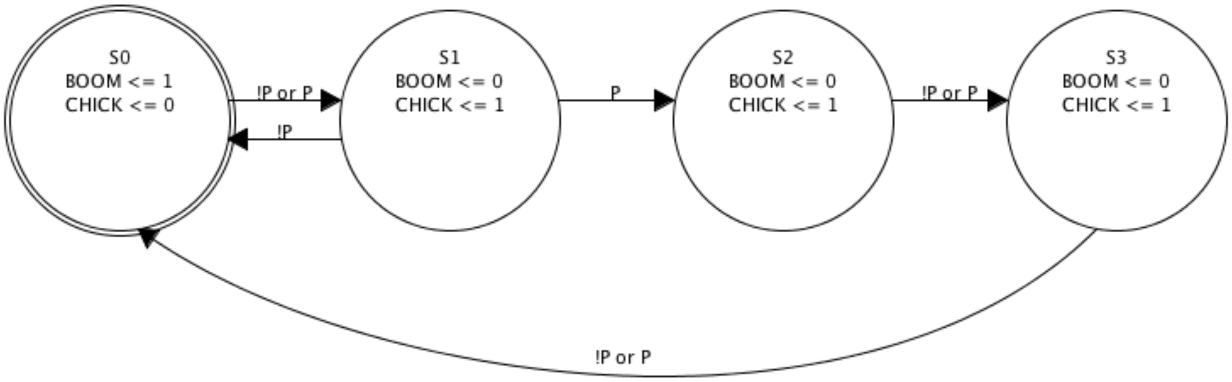
	INPUTS		OUTPUTS	
STATE NAME	Q1	Q0	BOOM	CHICK
S0	0	0	1	0
S1	0	1	0	1
S2	1	0	0	1
S3	1	1	0	1

This results in equations:

BOOM = Q1_ * Q0_

CHICK = Q1 + Q0

Now, let’s take a look at the new truth table for the state transitions. **Remember** that in the state diagram P means P = 1 and !P means P = 0. I’ll redraw the state machine diagram here for easy reference.



	INPUTS			OUTPUTS	
STATE TRANSITION	**Q1**	**Q0**	**P**	**D1**	**D0**
S0 —> S1	0	0	0	0	1
S0 —> S1	0	0	1	0	1
S1 —> S0	0	1	0	0	0
S1 —> S2	0	1	1	1	0
S2 —> S3	1	0	0	1	1
S2 —> S3	1	0	1	1	1
S3 —> S0	1	1	0	0	0
S3 —> S0	1	1	1	0	0

This table is more complex of course because now each state has two transitions leaving it. The diagram shows both arrows leaving S0 as one arrow with two conditions (!P or P). The diagram shows the two arrows leaving S1 very clearly. S2 and S3 also combine the two arrows into one with the conditions !P or P.

The K-maps and equations are:

Q1 \ Q0 P	00	01	11	10
0	0	0	1	0
1	1	1	0	0

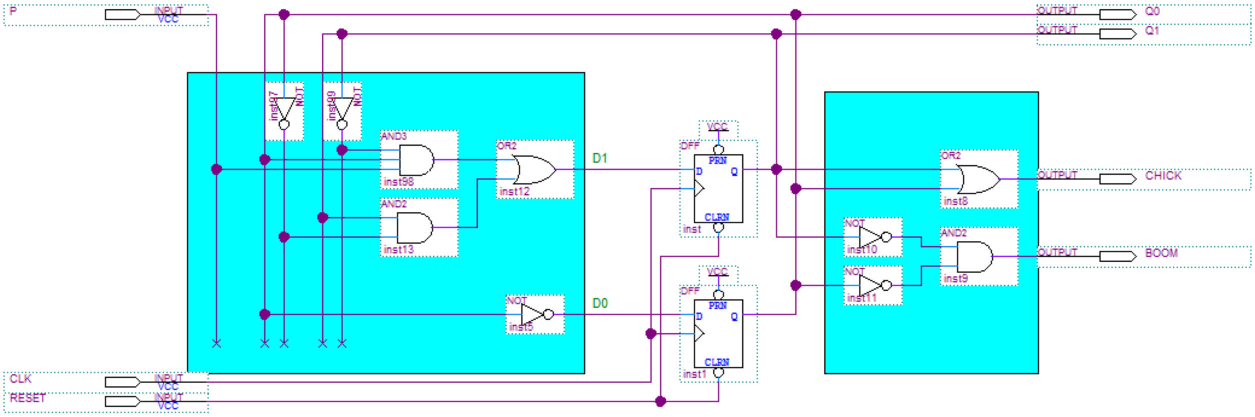
Q1 \ Q0 P	00	01	11	10
0	1	1	0	0
1	1	1	0	0

$$D1 = \overline{Q1} \bullet Q0 \bullet P + Q1 \bullet \overline{Q0}$$

$$D0 = \overline{Q0}$$

STEP 5: **DRAW THE CIRCUIT ON PAPER OR IN A CAD TOOL**

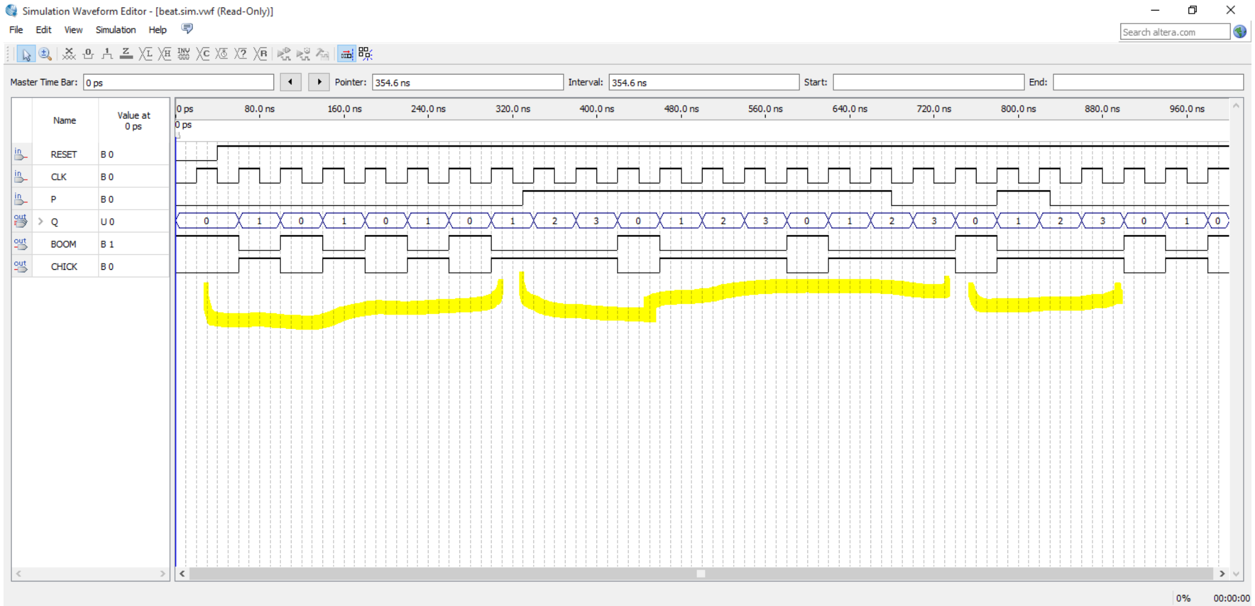
Let's use Quartus as the CAD tool for schematic and simulation. The Quartus schematic is shown below. I've shaded each combinational logic block in the Moore Machine Model to help you visualize them more easily.



Simulation of the circuit is a little more complex. The simulation has to verify that:

- P=0 produces boom-chick-boom-chick, boom-chick-boom-chick patterns
- P=1 produces boom-chick-chick-chick, boom-chick-chick-chick patterns
- And a change of P=1 to P=0 waits until S0 to adjust the beat pattern

The simulation diagram shows the machine behaving correctly.



This completes the design exercises for today.

HOMEWORK

Now, complete this homework in the rest of the Microsoft Word document that you started at the beginning of this lecture. Finish your work and email to me in PDF format by the end of the next academic day. **Use** the Print—> CutePDF Printer option to convert Microsoft Word to PDF.

1. **Design** a **new beat machine** that makes two patterns based on a percussionist foot pedal placed near the acoustic drum set. The pedal produces the machine input P. The patterns are:

   P = 0: Boom-chick-chick-chick
   P = 1: Boom-chick-chick-boom

2. **Draw** and **simulate** your solution in Quartus. **Ensure** that your simulation tests both rhythm patterns.
3. **Note** that your Microsoft Word document should include truth tables (use insert table), K-map tables and the derived equations (try to show K-map groups by shading table cells) for D1, D0, BOOM, and CHICK. The document should also include the Quartus schematic and simulation waveform diagrams. **Review** the "Printing Simulation Diagram" information on the course website under the Design Software link if needed.