Design of a Start-Up Sequence Controller for a Mammography Machine

Rakshak Udupa T S[1], Shashank K Holla[2*] and Namita Palecha[3].
[1]Dept of Electronics and Communication, RV College of Engineering, India,
[2]Dept of Electronics and Communication, RV College of Engineering, India,
[3]Dept of Electronics and Communication, RV College of Engineering, India
[1]rakshaktsudupa@gmail.com, [2]shashankholla@outlook.in, [3]namitapalecha@rvce.edu.in

Abstract: *Mammography, which is also calledMastography, is the process of using low-energy X-rays to inspect the human breast for screening and diagnostics. The purpose of mammography is to detect breast cancer early, usually by looking for specific lumps or microcalcifications. The X-rays used are usually around 30 kVp. Excessive voltage to such a machine would be harmful to the patient. Proper monitoring of temperature and pressure needs to be ensured. To ensure this, a start-up sequence module is developed. The start-up sequence module reads the digitised voltage, pressure and temperature reading from the sensor and asserts all the outputs to ensure that the machine is ready. The scan chain is formed of 13 scan flip-flops in this configuration. The synthesis mapped the design to 484 instances of cells in the opensource PDK technology. The design had a total area of 594 μm^2, with a cell width of 0.297 μm, and a height of 0.99 μm.*

Keywords:Mastography, Mammography, Start-Up Sequence Controller, PDK, YoSYS, Qflow

1.Introduction

1 in every 9 females and 1 in every 1000 males in India are observed to develop breast cancer every year [1]. With the developing methods to detect and cure breast cancer at an early stage, the number of patients is expected to reduce with the years to come. One of the essential and effective diagnostic methods for breast cancer is mammography. It helps in the early detection of cancer by analysing the tissue for characteristic masses that might be a sign of the disease. Screening mammography is a form of breast imaging that employs low-dose x-rays to diagnose cancer early on before signs appear when it is most treatable. A mammography device is a rectangular box that contains the tube that produces x-rays.[2]. The unit is only used for breast x-ray tests, with special attachments allowing only the breast to be exposed to the x-rays. A system is attached to the unit that holds and compresses the breast and places it to take photographs from various angles. The mammography procedure involves compressing the breast using parallel plates [3]. Breast tissues are evened out due to this, thus increasing image quality by reducing the tissue thickness that X-rays must penetrate. X-rays are a form of radiation to which many objects, including the human body, are transparent. An x-ray machine releases a tiny burst of radiation that passes through the body, capturing an image on photographic film or a special detector after it is carefully directed at the part of the body being examined. The x-rays are absorbed in varying amounts by different parts of the body. Soft tissue, such as muscle, fat, and organs, allows more x-rays to pass through them, while dense bone absorbs much radiation. In an x-ray, bones look white, and tissuesseem dark. The photos will be analysed by a radiologist, a physician specially qualified to supervise and interpret radiology exams. Any electronic system that involves taking in inputs, doing some process and giving out an output can be modelled as a state machine. This machine usually follows a particular set of steps before the system is ready to use. These steps are called the start-up sequence. This start-up sequence may include setting up the internal states, verifying that the input sources are correct and so on. They might also include Power-On Self-Test (POST) to check if the on-chip/on-board parts are working as expected.A state machine can be used to describe any electronic system that involves taking in inputs, performing a process, and producing an output. Before the system is ready to use, this machine goes typically through a set of processes. The start-up sequence refers to these phases. Setting up the internal states, ensuring that the input sources are correct, and so on are all possible parts of the start-up sequence.

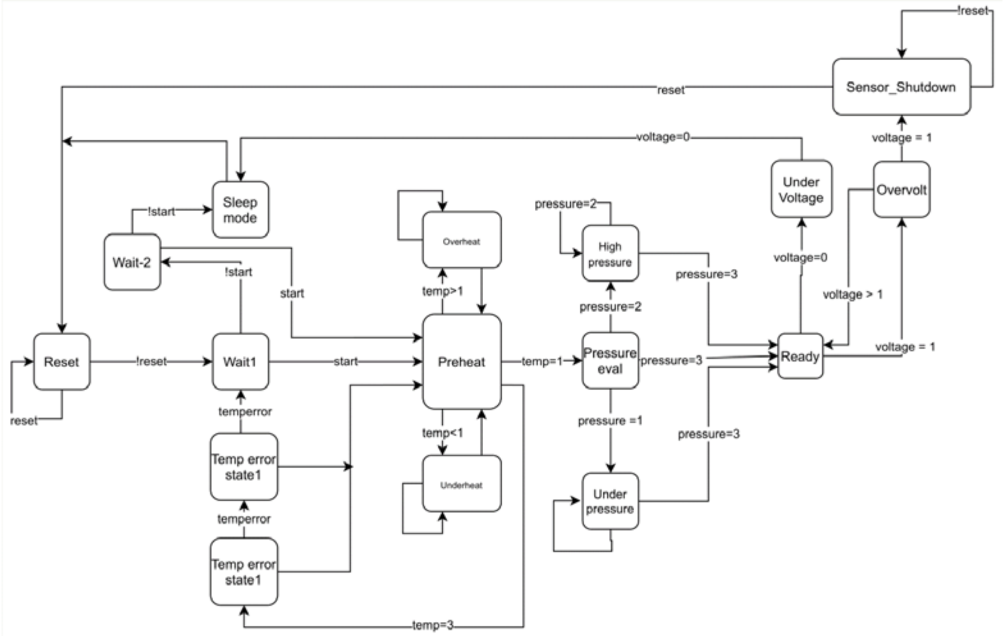


Figure 1 Start-up Sequence controller FSM

2. Designing the Finite State Machine

By understanding the working of the mammography machine, a model start-up sequence controller can be designed to test the BIST designed in the subsequent chapters. The start-up sequence controller is a finite state machine that puts the mammography machine into the ready state. The start-up sequence controller will need mainly 3 inputs from different sensors to get the mammography machine into ready state. They are:

- Temperature sensor: A temperature sensor must return the temperature of the electron gun to check if it is at the right temperature to generate X-rays.
- Pressure sensor: A pressure transducer to return the pressure values from the pneumatics and hydraulics.
- Voltage sensor: System input voltage to detect either undervoltage to put the system to sleep or overvoltage to turn off the system and prevent damage to sensors

The start-up sequence controller has 16 states. These states will be used to evaluate if the system's pressure and temperature values are in the ranges expected. The controller also has to evaluate the input voltage values and put the system to sleep in cases of undervolt, and shutdown sensors in cases of overvoltage.

The system takes in the following inputs

- Reset: 1-bit reset input
- Start: 1-bit start input
- Pressure: 2-bit input from pressure sensor
- Temperature: 2-bit input from temperature sensor
- Voltage: 2-bit input from voltage sensor

The flow of the finite state machine is shown in Figure 1. The states are explained in Table 1. Each state is assigned a state number. The input and the current state determine the output and the next state.

To implement the system, a state table was generated first doing state assignment. The state table has 4 columns: Present state, input, next state and output. Each present state and input combination will lead to a specific next state and output. The output for each state is explained in Table 2. The 3-bit output value helps the other subsystems determine the state of the start-up sequence controller. The other subsystems will wait until all bits of the output are HIGH, i.e., 111.

Table 1 Different states of the state machine

'RESET'	System stays in reset as long as RESET input is asserted. Reset is an asynchronous input and thus system can return to RESET from any state
'WAIT1'	System always goes to WAIT1 from RESET and waits for START to be asserted.
'WAIT2'	System enters WAIT2 from WAIT1 if START was not asserted in WAIT1
'READY'	System stays in READY state if Pressure, Voltage and Temperature sensors return correct values
'P_EVAL'	System enters P_EVAL state from PREHEAT if TEMP was maintained at expected value. If 2bit pressure sensor input has LSB high (11 or 01), then the system goes to READY. If the pressure sensor reports 10, then the system enters HIGH_P. If pressure sensor reports 00, then system enters LOW_P
'HIGH_P'	System waits here until the pressure sensor reports a value < 1, else returns to P_EVAL.
'LOW_P'	System waits here until the pressure sensor reports a value < 1, else returns to P_EVAL.
'PREHEAT'	System enters the PREHEAT state from WAIT1 or WAIT2 if START was asserted. Evaluation of temperature input is done here. If a temperature sensor gives an input of 11, then the system enters TEMP_ERROR1 state, implying some problem in temperature reading.
'OVERHEAT'	If the two-bit temperature sensor input is greater than 2'b01, then HIGH_P state is entered. System output will drive the heating unit to reduce temperature
'UNDERHEAT'	If the two-bit temperature sensor input is lesser than 2'b01, then HIGH_P is entered. System output will drive the heating unit to increase temperature
'TEMP_ERROR1'	Enters TEMP_ERROR2 if temperature reported is still 11, else returns to PREHEAT
'TEMP_ERROR2'	Enters WAIT state and restarts evaluation if reported temperature is still 11, else returns to PREHEAT
'OVERVOLT'	If input voltage sensor reading is 01, then system enters OVERVOLT state to subsequently enter SENSOR_SHUTDOWN
'UNDERVOLT'	If input voltage sensor reading is 00, then the system enters UNDERVOLT state to later enter SLEEP.

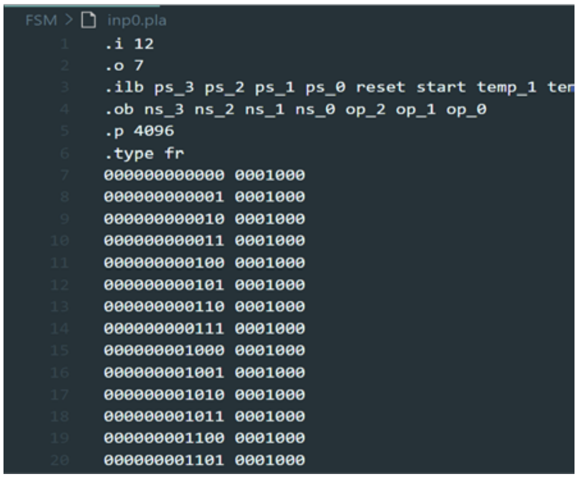
'SENSOR_SHUTDOWN'	System turns off supply to sensors to prevent damage. System stays in SENSOR_SHUTDOWN until RESET is asserted
'SLEEP'	System enters SLEEP when the UNDERVOLT state is reached. Stays in SLEEP until the voltage sensor reads > 0.

Table 2 Output pin description

Output pin value	Description
000	RESET
001	WAIT
010	TEMP_ERROR, SHUTDOWN
011	OVER VOLT
100	HEATING STAGE
101	PRESSURE EVAL STAGE
110	UNDERVOLT, SLEEP
111	READY

3.Implementation of State Table in Verilog

The process of implementing the FSM in Verilog is shown in Figure 2. A python script was written to generate the state table for all combinations of input and present states. The generated table was formatted in a specific format specified by BooM [5]. BooM is a Boolean minimization tool [22]. It expects a .pla file in a specific format as shown in Figure 3. The first 6 lines of the BooM configuration file defines the input types specified. There are 12 inputs and 7 outputs. The inputs are a combination of system inputs and present state values. There are 2 control inputs (START and RESET), 3 sensor inputs of 2 bits each (pressure, temperature and voltage), and 4-bit present state value. Thus 212 leads to 4096 combinations. The right-hand side of the column from line 7 specifies the outputs and the next state. The BooM tool generates a combinational Verilog code as output that can drive the sequential logic modelled using D flip flops.



```
.i 12
.o 7
.ilb ps_3 ps_2 ps_1 ps_0 reset start temp_1 ter
.ob ns_3 ns_2 ns_1 ns_0 op_2 op_1 op_0
.p 4096
.type fr
000000000000 0001000
000000000001 0001000
000000000010 0001000
000000000011 0001000
000000000100 0001000
000000000101 0001000
000000000110 0001000
000000000111 0001000
000000001000 0001000
000000001001 0001000
000000001010 0001000
000000001011 0001000
000000001100 0001000
000000001101 0001000
```

Figure 3 BooM file for generating the Verilog code for FSM

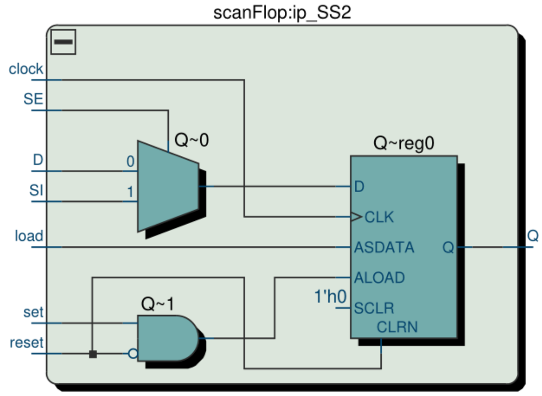


Figure 4 SynthesizedScan Flop

Integrate with scanFlops
Generate the verilog combo logic

Figure 2 Process of implementing the FSM

Theflip-flops in this design are replaced by scan flops to build a scan chain. This scan chain can be used to design a Built-In Self-Test (BIST) circuit. The scanflop has set, load, D, SI, SE, clock and reset as inputs. The set signal loads the load value onto the register. This is used in case there is a need to start the chain in a particular state. The SI is the Serial-In signal that is used when the flop is used in testing mode to shift in the values. The register loads in the value read from SI when SE (Scan-Enable) is high. Else, it loads in the D value. The synthesized scanFlop is shown in Figure 4. Once this flop is integrated with the combinational logic generated from BooM, the complete start-up sequence controller with scanflops is obtained. The synthesized result is shown in Figure 5.

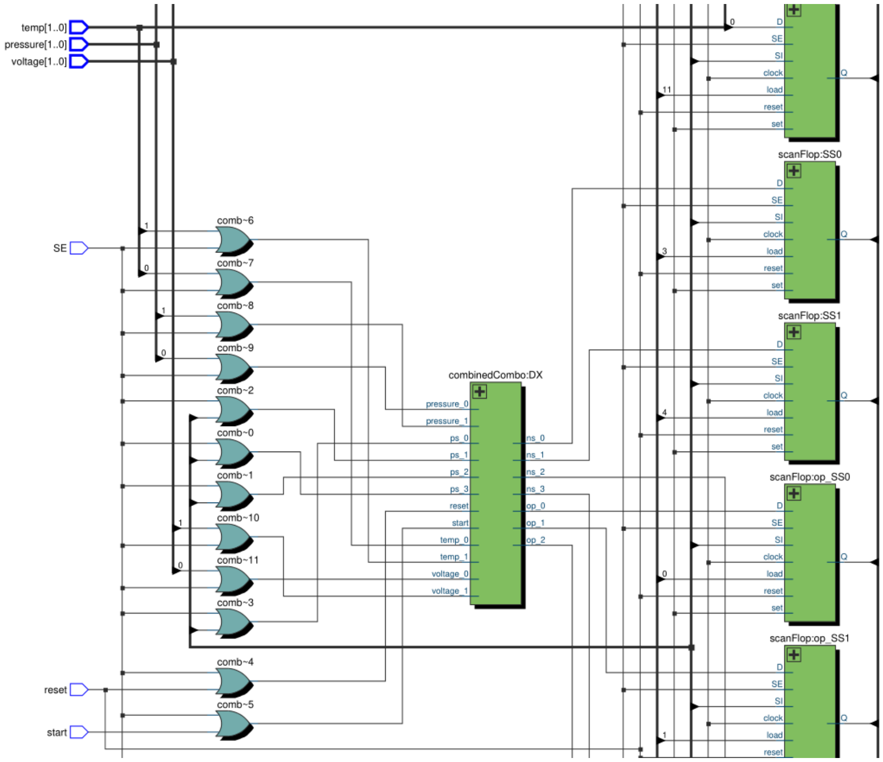


Figure 5 A sectional view of the synthesized start-up sequence controller

4.Results and Discussions

A functional simulation was done by checking if the system transitions from each state as expected. In Figure 6, it can be seen that the system senses a low pressure and enters the LOW P state. At time 100ns, the pressure sensor reports the correct pressure and hence returns back to P EVAL state. It then proceeds with the pressure evaluation and goes into READY. In Figure 7, it can be seen that the system successfully evaluates the temperature and pressure.However, at the READY state, the system sees that the voltage is Undervolt and thus enters the UNDERVOLT state and subsequently enters SLEEP. In Figure 8, it can be seen that the system senses a temperature error as the temp was observed to be 11. The system goes into TEMP ERROR1 state and subsequently TEMP ERROR2 state. Since the temperature error was not resolved even in TEMP ERROR2 state, the system returns to WAIT1 and restarts temperature evaluation.

The start-up sequence controller was synthesized using PDK in Yosys [6]. The synthesis results are tabulated in Table 3 and Table 4. Table 3 shows that of 13 unique cells were used from the library and a total of 484 instances were there. Table 4 summarizes the area utilized by the design.

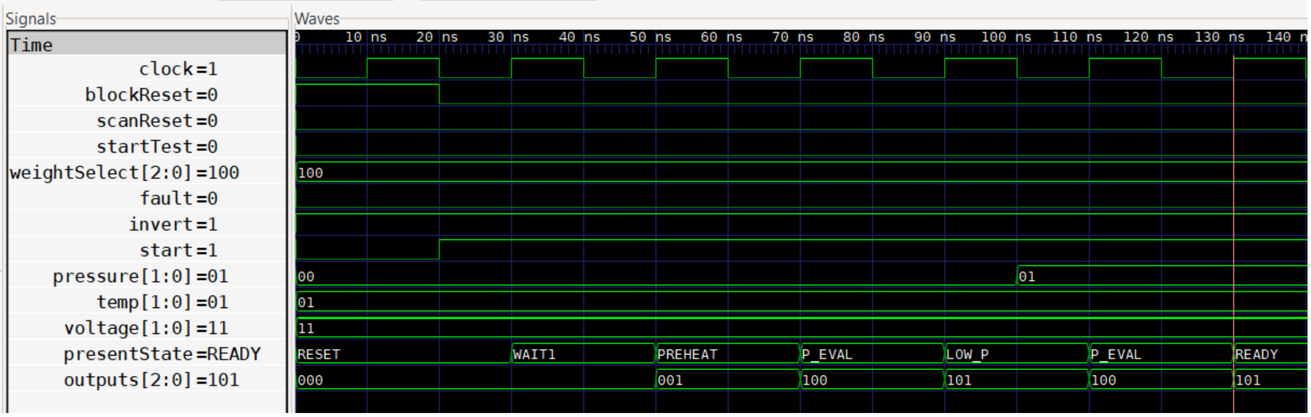


Figure 6 After successfully evaluating pressure, temperature and voltage, system enters

.

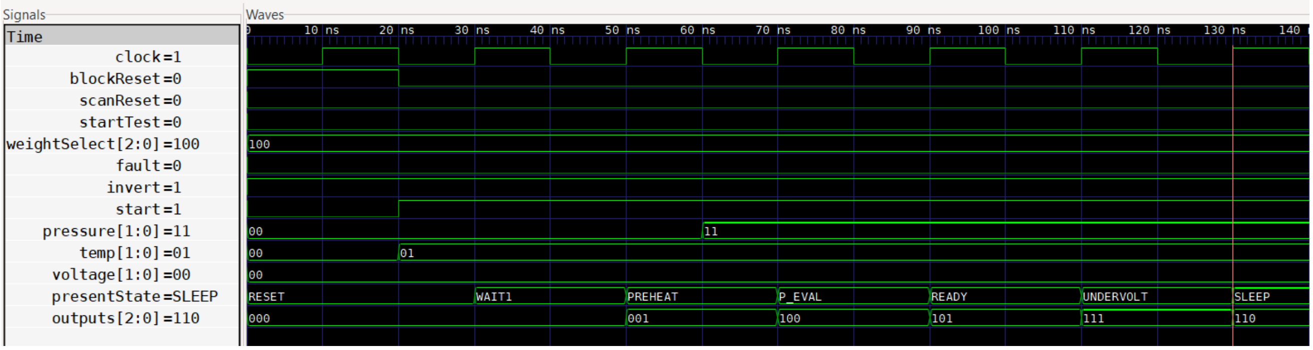


Figure 7System enters sleep since undervolt state was detected

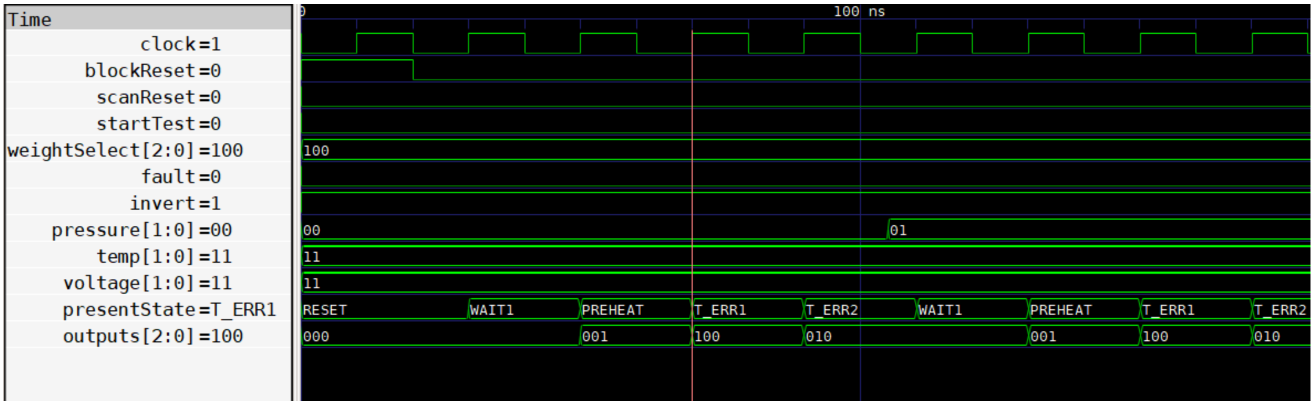


Figure 8 Invalid temperature sensor inputs lead to the system entering the T_ERR

Table 3 Library cells used in the design

Library Cell	Number of instances
AND2X2	6
AOI21X1	1
BUFX2	273
DFFSR	12
NOR3X1	10
INVX1	29
NAND2X1	40
NAND3X1	40
NOR2X1	22
OAI21X1	34
OAI22X1	2
OR2X2	13
XNOR2X1	1

Table 4 Placement log

Dimension	Value
Total cell width	0.297µm
Total cell height	0.99µm
Total cell area	594µm^2
Total core area	594µm^2
Average cell height	2 nm

5.Conclusion

The start-up sequence controller works as expected. When the temperature is high, the start-up sequence goes to the temp_err state. Until sufficient (not too high nor too low) amount of pressure and voltage is sensed, the start-up sequence controller does not assert all output signals to reach the ready state. The start-up sequence controller transits to error states whenever invalid sensor readings are detected. The design uses 13 scan flip-flops to form the scan chain. After synthesising the design using opensource PDK, the design used 484 instances of library cells and 211 instances excluding BUF cells. The design had a cell width of 0.297µm, height of 0.99 µm and a total area of 594 µm^2

References

[1]. P. Mathur, K. Sathishkumar, M. Chaturvedi, P. Das, K. L. Sudarshan, S. Santhappan, V. Nallasamy, A. John, S. Narasimhan, and F. S. a. Roselind, “Cancer statistics, 2020: Report from national cancer registry programme, india,” JCO Global Oncology, no. 6, pp. 1063–1075, 2020, PMID: 32673076. doi: 10.1200/GO.20.00122.

[2]. M. Ebrahimi, “Breast imaging: Mammography, digital tomosynthesis, dynamic contrast enhancement,” in. Jan. 2017, isbn: 9780128012383. doi: 10.1016 /B978-0-12-801238-3.99943-4.

[3]. P. Raghavendra and T. Pullaiah, “Biomedical imaging role in cellular and molecular diagnostics,” in. Jan. 2018, pp. 85–111, isbn: 9780128136799. doi: 10.1016 /B978-0-12-8136 79-9.00004-X

[4]. K. Mielcarek, A. Barkalov and L. Titarenko, "Designing Moore FSM with unstandard representation of state codes," *2016 5th International Conference on Modern Circuits and Systems Technologies (MOCAST)*, 2016, pp. 1-4, doi: 10.1109/MOCAST.2016.7495106.

[5]. J. Hlavicka and P. Fiser, "BOOM-a heuristic Boolean minimizer," IEEE/ACM International Conference on Computer Aided Design. ICCAD 2001. IEEE/ACM Digest of Technical Papers (Cat. No.01CH37281), 2001, pp. 439-442, doi: 10.1109/ICCAD.2001.968667.

[6]. D. Shah, E. Hung, C. Wolf, S. Bazanski, D. Gisselquist and M. Milanovic, "Yosys+nextpnr: An Open Source Framework from Verilog to Bitstream for Commercial FPGAs," 2019 IEEE 27th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM), 2019, pp. 1-4, doi: 10.1109/FCCM.2019.00010.