An IEEE-Compatible JTAG Test-Access-Port Controller for RSFQ Logic and Systems

Ivan Sutherland, Quinn Morgan, Warren A. Hunt Jr., Vivek Ramanathan, and Marly Roncken, *Member, IEEE*

***Abstract*—JTAG, an IEEE-standard test protocol, is widely used for testing commercial and experimental CMOS chips but appears unused for testing superconducting chips. The test-access-port (TAP) controller described here facilitates the use of JTAG to reduce the cost – in time and effort – of testing superconductive circuits. Our RSFQ-based TAP controller shifts data serially through a scan chain that transfers data in parallel to and from a circuit under test. Suitably designed scan chains can provide initial values to superconducting circuits and retrieve results to explore circuit behavior. A control computer or an internet connection can operate our TAP controller through a simple five-wire interface. Our TAP controller implements the TAP part of IEEE standard 1149.8.1-2012 in an area of only 580 × 280 micrometers, less area than two bonding pads. Our TAP Controller design information and MIT SFQ5ee layout will be freely available.**

***Index Terms*—RSFQ logic, IEEE test-access controller (TAP), super-conducting logic.**

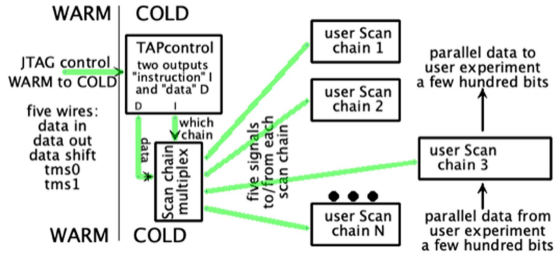

Fig. 1. An on-chip TAP controller is the heart of a JTAG system. A serial interface of only five wires controls the internal state of the TAP controller. Depending on its internal state, the TAP controller connects the serial interface to either a serial data port, D, or an *instruction* shift register, I. The instruction shift register tells the scan chain multiplexer which of several scan chains to connect to the data port for serial data input and output. The broad green lines in this figure represent connections for multiple pulse signals on JTLs or in short wires, if possible.

I. Introduction

TO DETERMINE if an integrated circuit works properly, one must initialize the circuit, operate the circuit, and retrieve its final state for examination. Because external chip I/O connections are expensive in every technology, test procedures for both commercial CMOS chips and superconducting chips use serial transfer to put test data onto and retrieve test results from a chip under test. Designers of superconducting chips serialize data in diverse ways, each designer building a different unique form [1]. In contrast, nearly all experimental and commercial CMOS chips use the IEEE standard JTAG test interface [2], [3]. Its simplicity, widespread use, modest speed requirement, flexibility, and simple five-wire interface (Fig. 1) make JTAG attractive for use with experimental superconducting chips. Including a test mechanism increases the value one can extract from an experimental chip. The JTAG interface is widely used because its five-wire interface minimizes the number of pins dedicated to finding manufacturing errors in commercial chips. We bring the IEEE 1149.8.1-2012 [4] test-access protocol to RSFQ [5], [6] circuits by offering the RSFQ Test Access Port (TAP) controller described here.

Manuscript received 7 November 2022; revised 21 January 2023; accepted 20 February 2023. Date of publication 6 March 2023; date of current version 21 April 2023. This work was supported in part by Army contract W911NF-20-C-0032, administered by Joseph M. Coyle. *(Corresponding author: Warren A. Hunt.)*

The authors are with ForrestHunt, Inc., Austin, TX 78746 USA (e-mail: hunt@cs.utexas.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TASC.2023.3251942.

Digital Object Identifier 10.1109/TASC.2023.3251942

An ecosystem of devices, knowledge, and instructional material has evolved around the JTAG testing standard. Commercial JTAG drivers that connect to USB or internet ports are available for less than 400 US dollars [7]. We believe rapid development of superconducting technology will require remote access to suitably mounted and cooled chips so that designers can test their own creations without the cost and hazard of managing cryogenic equipment. Using JTAG would allow the superconductor electronics community to use all of the existing test tooling and machinery available to the CMOS electronics community. We hope to make the JTAG ecosystem available to the superconducting community.

II. The TAP Controller

Fig. 1 is a block diagram of on-chip parts for a JTAG test system. Five JTAG control wires from the warm outside environment drive the system. Serial **data in** and serial **data out** wires permit concurrent retrieval of former on-chip state and insertion of new initialization data.

Our TAP controller can put data into and retrieve data from on-chip serial *scan chains* of arbitrary lengths such as shown in Fig. 2. The D2Latches in each scan chain have two sets of control terminals. One set of terminals shifts data through the latches and the other set connects latches into the system. Using several short scan chains rather than a single longer scan chain may increase flexibility and reduce test time. Our TAP controller accommodates the IEEE standard mechanism [4] for selecting among multiple scan chains, although a scan-chain multiplexer remains a future endeavor.

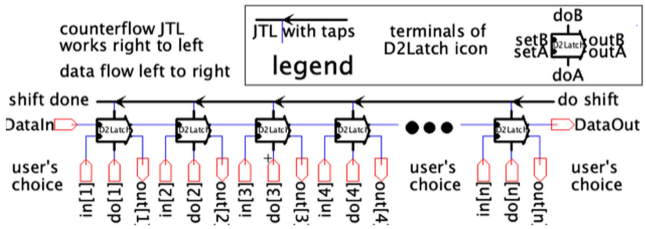

Fig. 2. A scan chain of D2Latches holds only one bit in each latch. One of the two sets of control terminals for each latch connects the D2Latches into a shift register; the other set of control terminals can serve some other purpose, for example to make a counter. Because the shift string shifts the actual count bits, shifting automatically initializes or retrieves the count value.

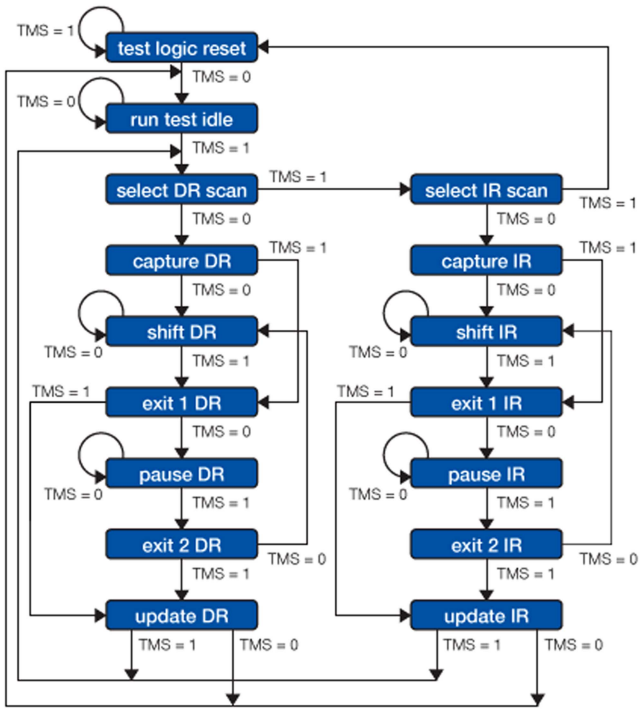

Fig. 3. State transition diagram for the IEEE TAP controller. A blue box represents each state. The **TMS=0** or **TMS=1** labels indicate which transitions result from each state-advancement (clock) signal. The **TMS=0** transitions out of the states **capture**, **shift**, and **update** invoke instruction and data scan chain actions to load, shift, and read external shift registers. This diagram is available publicly.

III. The TAP State Machine

A TAP controller is a state machine whose state transition diagram appears in Fig. 3. Our TAP controller state machine uses a one-hot configuration of 16 D2Latches [8], see Figs. 4 and 5, one for each state in Fig. 3. Only the one latch representing the present state is ever FULL; the rest of the latches are EMPTY. Two long JTLs pass timing signals called, **tms0** and **tms1**, through the **tms0in** and **tms1in** terminals of each and every latch. To change the state, these control JTLs follow the sequence given in Table I. Every EMPTY latch remains silent and EMPTY, but the one-and-only latch representing the present state, and therefore FULL, drains and emits a pulse at an output terminal. A connecting wire or short JTL between that output terminal and an input terminal of another latch passes the FULL state to the latch that represents the successor state. This state transition mechanism allows the state transition diagram of Fig. 3 to be the wiring diagram of the TAP controller, because each state transition arrow in Fig. 3 corresponds to a wire or short JTL between a sending state latch and a receiving state latch. A *Master Clear* signal is unnecessary because the IEEE specification includes a reset sequence: a long string of **tms1** transitions forces the state machine into the state **test logic reset** at the top of Fig. 3.

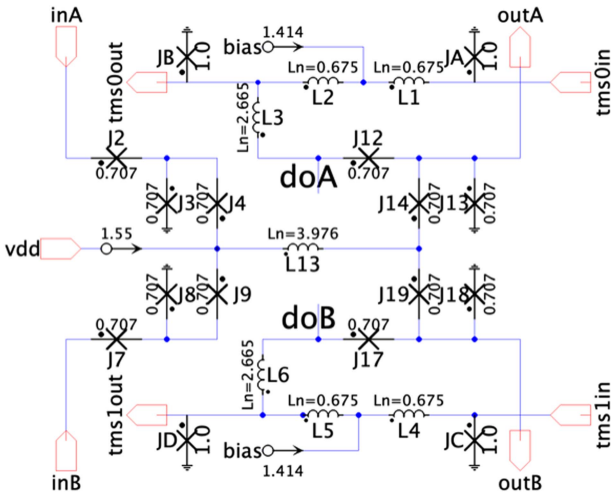

Fig. 4. Schematic of our D2Latch and parts of two **tms** JTLs. JJs from the **D2Latch** carry names **J2**–**J19**; JJs from the **tms** JTLs carry names **JA**, **JB**, **JC**, and **JD**. The fork to drive the D2Latch is built in because **JB** drives both the **doA** terminal of this latch via L3 and the output terminal, **tms0out**, connected to the rest of the **tms** JTL. In our layout, Fig. 5, the JTL terminals **tms0in**, **tms1in**, **tms0out**, and **tms1out** align with gaps between JJs to facilitate access (from above and below) to terminals **inA**, **inB**, **outA**, and **outB**. A scan chain of the latches uses one of the JTLs for counterflow shift; notice that data flow left-to-right, from **inA** to **outA**, but the control JTLs flow right-to-left. Component values are presented in PSCAN normalized inductance units, Ln = 2.632e-12 Henry, and normalized current units, In = 125e-6 Amp. Horizontal arrows represent bias current resistors chosen to provide the normalized bias currents shown. Separate bias supplies, **bias** (for JTLs) and **vdd** (for latches), facilitate electrical margin measurements.

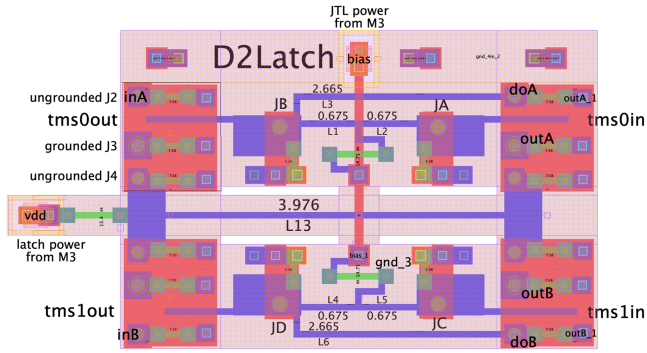

Fig. 5. Layout of our D2Latch and parts of two **tms** JTLs. In this layout, sets of three JJs, like **J2**, **J3** and **J4** (upper left) share **M5** metal squares (red) as their common connection. The bypass resistor (green) for each JJ lies next it. A gap between the outer two JJs accommodates the JTL connection that passes from latch-to-latch near terminals **tms0in**, **tms1in**, **tms0out**, and **tms1out**. Corner data terminals, **inA**, **inB**, **outA** and **outB** remain unobstructed because they are outside the horizontal metal **M6** (blue) JTL connection. Our layout tool estimates and annotates wire inductance dynamically as layout changes. Its name and estimated inductance value appear near many wires; for example, the center **M6** (blue) horizontal quantizing inductor **L13** shows the estimated normalized inductance value 3.976. The central vertical metal **M5** (red) wire and two bias resistors (green) provide JTL bias via coupling inductors **L1**, **L2**, **L4**, and **L5**.

TABLE I
STATE ADVANCEMENT SEQUENCE FOR TMS INPUTS

Sequence for **tms0**	Sequence for **tms1**
run_test_idle	test_logic_reset
update_DR	select_IR_scan
pause_DR	select_DR_scan
exit_1_DR	run_test_idle
shift_DR	update_DR
exit_2_DR	exit_2_DR
capture_DR	pause_DR
select_DR_scan	exit_1_DR
test_logic_reset	shift_DR
update_IR	capture_DR
pause_IR	update_IR
exit_1_IR	exit_2_IR
shift_IR	pause_IR
exit_2_IR	exit_1_IR
capture_IR	shift_IR
select_IR_scan	capture_IR

Our TAP Controller uses counterflow clocking [9] to distribute **tms0** and **tms1** pulses to its latches. Counterflow clocking ensures that each and every state transition is atomic because the destination latch of every state transition precedes its source latch in the control JTL sequence and therefore receives the control pulse before its source can emit its transition pulse. Other control pulse delivery sequences risk allowing a single control pulse to advance state twice. It is fortunate that the state transition diagram of Fig. 3 permits counterflow clocking for both **tms0** and **tms1** pulses. Our layout capitalizes on the symmetry evident in Fig. 3 and positions the state latches to minimize the physical lengths of state-transition JTLs and wires as well as the physical lengths of the JTLs that deliver **tms0** and **tms1** pulses.

Table I lists the counterflow delivery sequences for **tms0** and **tms1** pulses. The **tms0** and **tms1** pulses pass through separate long JTLs to drain the successor latch of each and every state transition **before** emitting the content (if FULL) of that transition's source latch. In effect, these long control-propagating JTLs wander among the latches, reaching each in the sequence given in Table I.

IV. THE D2LATCH

For our TAP controller, we use the two-input, two-output **D2Latch** of Figs. 4 and 5. The **D2Latch** accepts a data input pulse on either of two inputs, **inA** or **inB**, to *fill* its state-holding or *quantizing* inductor, **L13**. If the latch is FULL, the latch emits an RSFQ pulse at **outA** in response to a read pulse at **doA** and becomes EMPTY. Similarly, if the latch is FULL, the latch emits an RSFQ pulse at **outB** in response to a read pulse at **doB** and becomes EMPTY. In contrast, if the latch is EMPTY, the latch ignores read pulses at **doA** or **doB** and remains silent, emits nothing, and remains EMPTY. Pulses at **doA** and **doB** must be separated in time.

The **D2Latch** emits its stored bit at one of its two output terminals according to which of its two **do** terminals receives the read pulse. We use the name "do" to distinguish control of RSFQ output timing, which tells when to **emit** an output pulse, from control of CMOS input timing, where we use the name "clock" for a signal that tells when to **capture** fresh data. Pulse logic can **emit**, but can never **capture**, because to capture requires a previously-set level signal, and level signals are absent from pulse logic.

V. PHYSICAL ORGANIZATION OF OUR TAP CONTROLLER

For our TAP controller, D2Latches with two data outputs suffice because each of the 16 states in Fig. 3 has at most two output state transitions. Two inputs also suffice for most states, but an extra merge module accommodates states **shiftDR** and **shiftIR** that require three inputs.

Our TAP controller layout (not shown) has three parts that capitalize on the symmetry of Fig. 3. A middle part holds the state latches for the four topmost states of Fig. 3. A side part, used twice, folds the six state latches for one of the two identical lower columns of Fig. 3 into two rows with **capture**, **shift**, and **exit1** in an upper row and **pause**, **exit2**, and **update** in a lower row. Two copies of the side part flank the middle part to make a two-row layout eight latches wide. This arrangement minimizes the lengths of both the latch-to-latch state transition wires or JTLs and the long **tms0** and **tms1** control JTLs. The five JTAG control wires that come from a chip's bonding pads attach to the bottom row while scan-chain controls emerge from the top row. The TAP controller conveniently divides the wires from control pads (below) from the connections to scan chains (above).

VI. USE OF THE TAP CONTROLLER

Control inputs pulses, **tms0** and **tms1**, change the state of the TAP controller. In the six states, **capture {IR,DR}**, **shift {IR,DR}**, and **update {IR,DR}**, **tms0** pulses also drive shift registers. Two shift registers attached directly to the TAP controller drive a scan-chain multiplexer (not described here) that attaches one of several user scan chains to the TAP controller. Scan chains that use counterflow clocking may be of any length because they return a completion pulse once they finish their actions.

The scan chain multiplexer decodes the state of the instruction scan chain to attach one of several data scan chains to the data chain of the TAP controller. Multiple data scan chains can accommodate entirely different on-chip experiments, or serve to initiate and retrieve additional state bits from complex logic. Because changing the state of the TAP controller may be required and because multiple scan chains complicate layout, users may initially prefer to forgo the multiple data chain capability of JTAG, omit the scan chain multiplexer, and drive a single data scan chain directly from the TAP controller.

VII. SIMULATION/OPERATION OF TAP CONTROLLER

Fig. 6 shows results of a simulation of the whole TAP controller to demonstrate that it cycles properly though the states shown in the left column of Fig. 3. The top panel shows the primary input **tms0** and **tms1** phase changes that drive the simulation. The sequence **tms1**, **tms0**, **tms0**, **tms1**, followed by **tms0**, **tms1**, **tms1**, **tms0** matches the state-transition labels for a circular tour down the left column of Fig. 3. The bottom panel shows the currents in the quantizing inductors of the **D2Latches** for the corresponding states. A separate simulation (not shown) of the **D2Latch** of Fig. 4 demonstrated that it rejects redundant **input** and **do** pulses.

Our TAP controller implements the core part of the IEEE 1149.8.1-2012 specification in only 580 × 280 micrometers in size, less than the area of two bonding pads. This is a tiny fraction of a chip: about 0.65% of 0.5 cm square chip.

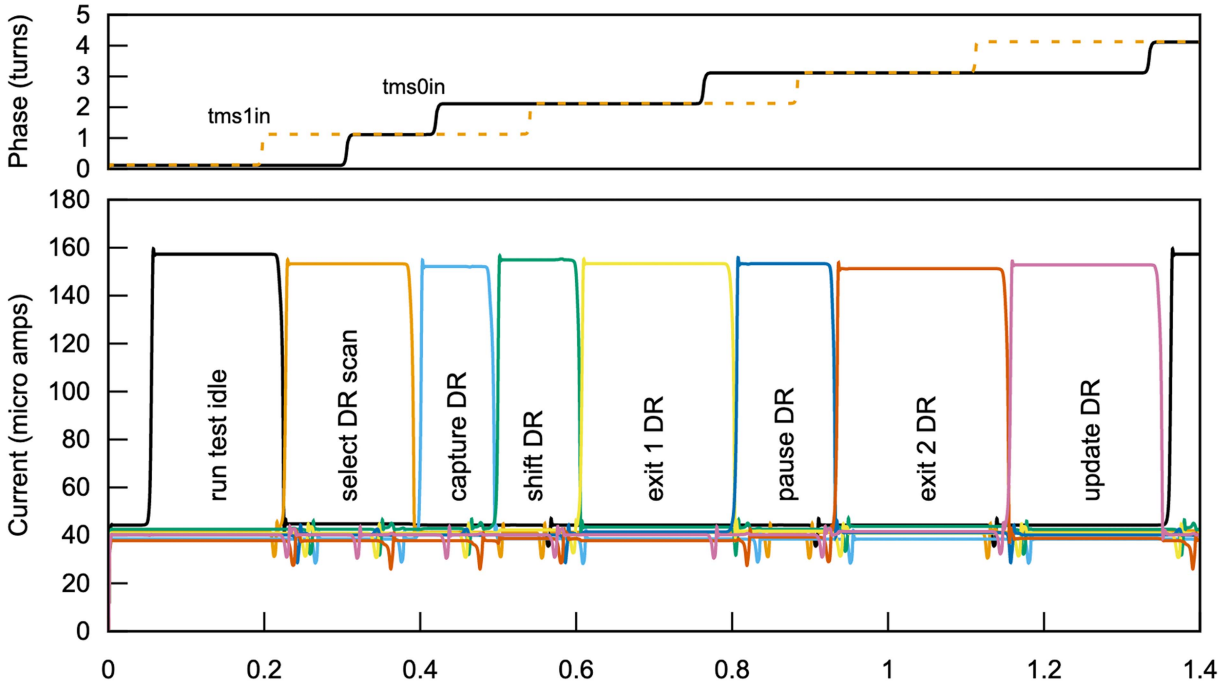

Fig. 6. Timing diagrams for a simulation of TAP controller. The top panel shows signals that trigger simulation actions as phase changes **tms** signals. Computer or internet connections from the warm outside environment will deliver such signals to the TAP controller at a much more leasurely pace than shown here. The simulation initially fills the **run test idle** with a control pulse not shown. The simulated sequence matches state-transition labels for a circular tour down the left column of Fig. 3: **tms1**, **tms0**, **tms0**, **tms1**, followed by **tms0**, **tms1**, **tms1**, **tms0**. The bottom panel shows the currents in quantizing inductor, **L13**, for each latch with the name of the state represented by that latch. The *one-hot* encoding used in our TAP controller is evident because only one latch at at a time has large **L13** current and each state change shows that a subsequent latch fills shortly after a control pulse drains its predecessor. Delays between events in the top panel and state changes in the bottom panel vary because different latches are at different positions in the **tms** JTL. The starting delay of different state transitions varies because some state transitions use a short wire but other layout distances require a few stages of JTL.

VIII. Conclusion

The superconducting electronics community can profit by automating the test and analysis of experimental circuits [10]. The present cost in time and effort of using a unique test setup for each chip is far too high. Our RSFQ TAP Controller provides the IEEE 1149.8.1-2012 standard interface for initialization, testing, and analysis of superconducting circuits. With this interface, a standard test harness may suffice for many chips – test engineers or designers themselves can use the a standard test harness on a superconducting chip as soon as the chip emerges from fabrication.

We are willing to share our design data and layouts as they become available. We look forward to engaging with the superconducting community to further the test capability on which progress depends.

Acknowledgment

The authors would like to thank our colleagues Coenrad Fourie, Evan Golden, Matt Guyton, Alex Kirichenko, Vasili Semenov, Glenn Shirley, Sergey Tolpygo, and Alex Wynn, for many useful discussions.

References

[1] A. F. Kirichenko, O. A. Mukhanov, and A. I. Ryzhikh, "Advanced on-chip test technology for RSFQ circuits," *IEEE Trans. Appl. Supercond.*, vol. 7, no. 2, pp. 3438–3441, Jun. 1997, doi: 10.1109/77.622124.
[2] JTAG Technologies. [Online]. Available: https://www.jtaglive.com
[3] XJLink2–USB JTAG Controller, Cambridge Technology Group. 2023. [Online]. Available: https://www.xjtag.com/products/hardware/xjlink-xjlink2-controller/
[4] *IEEE Standard for Boundary-Scan-Based Stimulus of Interconnections to Passive and/or Active Components*, IEEE Standard 1149.8.1-2012, Aug. 2012, doi: 10.1109/IEEESTD.2012.6259815.
[5] P. Bunyk, K. Likharev, and D. Zinoviev, "RSFQ technology: Physics and devices," *Int. J. High Speed Electron. Syst.*, vol. 11, no. 1, pp. 257–305, 2001, doi: 10.1142/S012915640100085X.
[6] K. K. Likharev and V. K. Semenov, "RSFQ logic/memory family: A new Josephson-junction technology for sub-terahertz-clock-frequency digital systems," *IEEE Trans. Appl. Supercond.*, vol. 1, no. 1, pp. 3–28, Mar. 1991, doi: 10.1109/77.80745.
[7] JTAG Live Controller. [Online]. Available: https://www.jtaglive.com/product/low-cost-usb-jtag-live-controller-interface/
[8] S. V. Polonsky, V. K. Semenov, and A. F. Kirichenko, "Single flux, quantum B flip-flop and its possible applications," *IEEE Trans. Appl. Supercond.*, vol. 4, no. 1, pp. 9–18, Mar. 1994, doi: 10.1109/77.273059.
[9] O. A. Mukhanov, "Rapid single flux quantum (RSFQ) shift register family," *IEEE Trans. Appl. Supercond.*, vol. 3, no. 1, pp. 2578–2581, Mar. 1993, doi: 10.1109/77.233529.
[10] M. Roncken, E. Esimai, V. Ramanathan, W. Hunt Jr., and I. Sutherland, "State access for RSFQ test and analysis," *IEEE Trans. Appl. Supercond.*, early access, Mar. 2, 2023, doi: 10.1109/TASC.2023.3251949.