

## 1 Introduction

This application note describes how to use the FlexIO module to emulate state machine. Using State mode enables the user to implement any state machine with up to eight states, eight outputs, and three selectable inputs per state. This feature allows basic control functions to be offloaded from the CPU, which could potentially remain in low power mode.

## 2 Development platform

In order to emulate the state machine, an i.MX RT1010 EVK board was used in this demo. In this application case, the used pins are as shown in [Figure 1](#).

### Contents

<b>1 Introduction.....</b>	<b>1</b>
<b>2 Development platform.....</b>	<b>1</b>
<b>3 FlexIO state mode emulation.....</b>	<b>3</b>
<b>4 Conclusion.....</b>	<b>6</b>
<b>5 References.....</b>	<b>6</b>



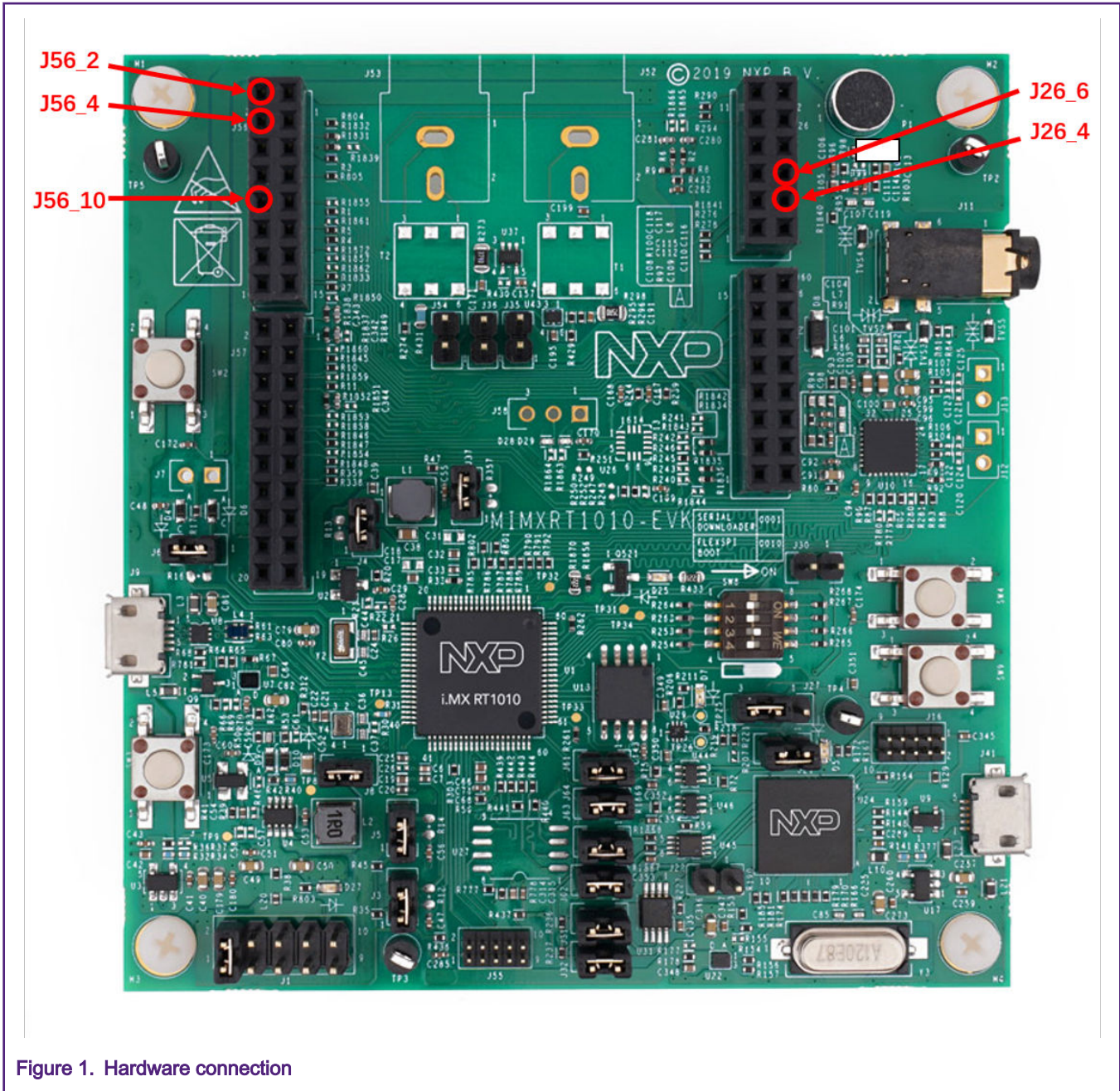


Figure 1. Hardware connection

On the i.MX RT1010, FLEXIO has a total of 27 pins. The user can use the state mode to achieve up to 8 states, each state controls up to eight pin outputs.

In this case, a total of three states are emulated. Each state enables three output pins, FXIO[0], FXIO[1], and FXIO[2]. These three state machines can be controlled by three optional input pins. The three pins selected in this application are FXIO[21], FXIO[22], and FXIO[23].

Table 1 lists the FlexIO pins used in this case:

Table 1. Used FlexIO pins

FlexIO pin	Pin location	Pin name	State
flexio1.FLEXIO00	J56-10	GPIO_08	Enable output

Table continues on the next page...

Table 1. Used FlexIO pins (continued)

FlexIO pin	Pin location	Pin name	State
flexio1.FLEXIO01	J56-02	GPIO_09	Enable output
flexio1.FLEXIO02	J56-04	GPIO_10	Enable output
flexio1.FLEXIO03	—	GPIO_11	Disable
flexio1.FLEXIO04	—	GPIO_12	Disable
flexio1.FLEXIO05	—	GPIO_13	Disable
flexio1.FLEXIO06	—	GPIO_SD_00	Disable
flexio1.FLEXIO07	—	GPIO_SD_01	Disable
flexio1.FLEXIO21	J26-04	GPIO_AD_09	Enable input
flexio1.FLEXIO22	J26-06	GPIO_AD_10	Enable input
flexio1.FLEXIO23	—	GPIO_AD_11	Enable input

To make application case work, take the following into consideration.

- Remove R792, and solder R800 with 0 Ω resistor on RT1010 EVK board.
- Remove jumper caps J31 and J32, remove resistors R237 and R236.
- Make sure GPIO\_AD\_11 is connected to the ground.
- Power on board with USB cable plugged to J41.

### 3 FlexIO state mode emulation

#### 3.1 State mode overview

The Shifter of the FlexIO module provides six working modes, and this application note focuses on its state mode. Set the state mode by writing the SMOD of FLEXIOx\_SHIFTCTLn register to 0x6. FlexIO state mode provides up to eight state controls. These controls allow offloading from the CPU, and the state mode function can remain in low power mode. Figure 2 provides a detailed view of the Shifter microarchitecture.

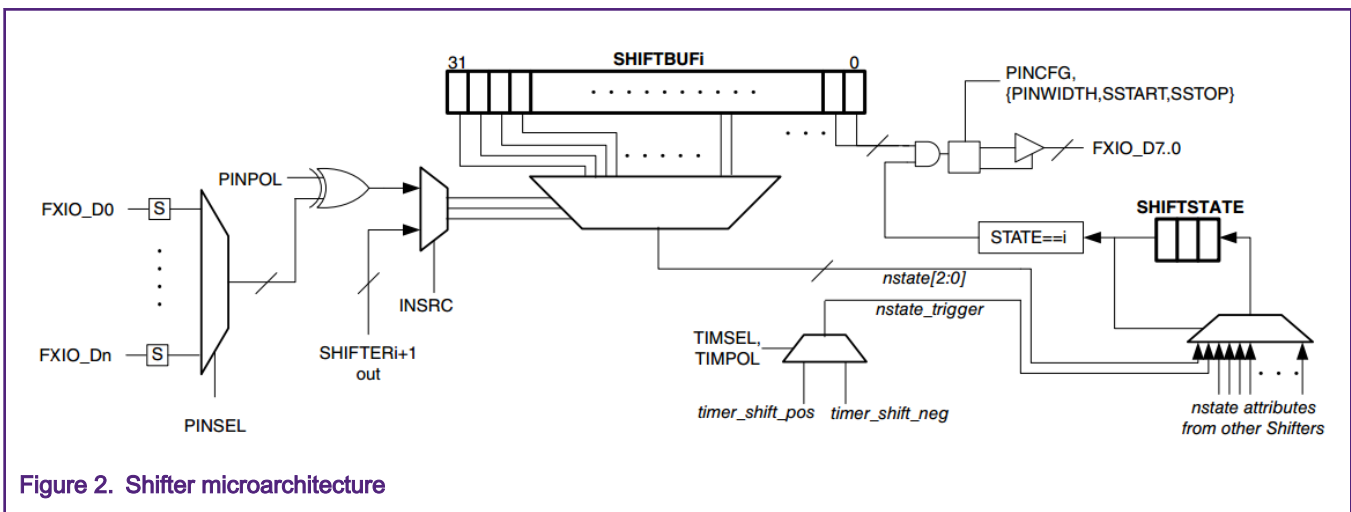


Figure 2. Shifter microarchitecture

Figure 3 shows the I/O assignments for a given state in general. The user can select three consecutive flexio pins as the control input of the state machine by setting the PINSEL of the register FLEXIOx.SHIFTCTLn. The three inputs to the state can also be

represented by three LSB bits of the next (n+1) Shifter. The eight outputs from the state are fixed and assigned to FlexIO pins, FXIO[0]-FXIO[7]. FXIO[0]-FXIO[7] can be enabled or disabled by setting PWIDTH [3: 0], SSTOP [1: 0], and SSTART [1: 0] of FLEXIOx.SHIFTCFGn register. In addition, the PINCFG of FLEXIOx.SHIFTCTLn register should be configured as output.

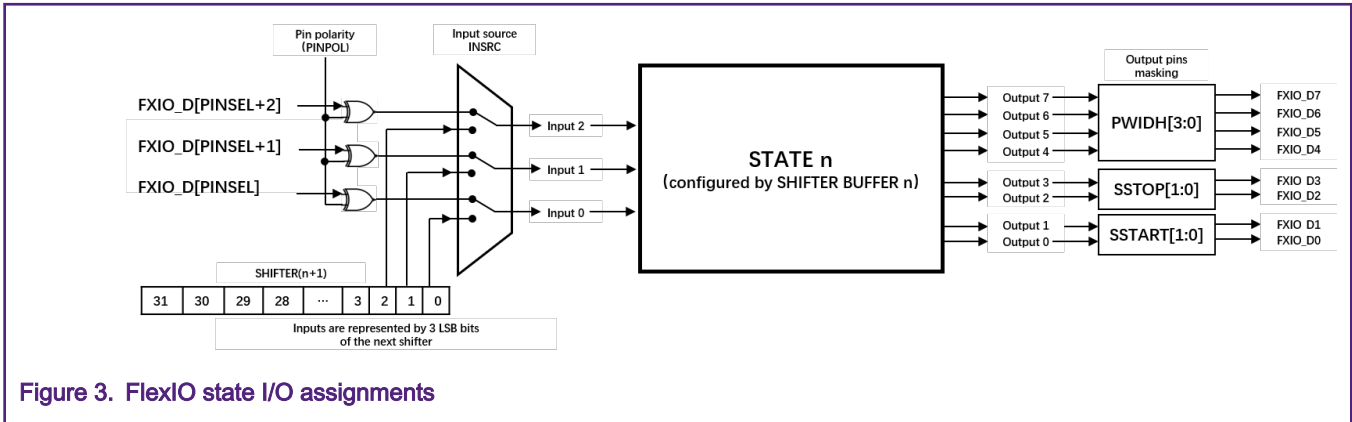


Figure 3. FlexIO state I/O assignments

### 3.2 State mode emulation

Figure 4 shows the state machine diagram in the application case.

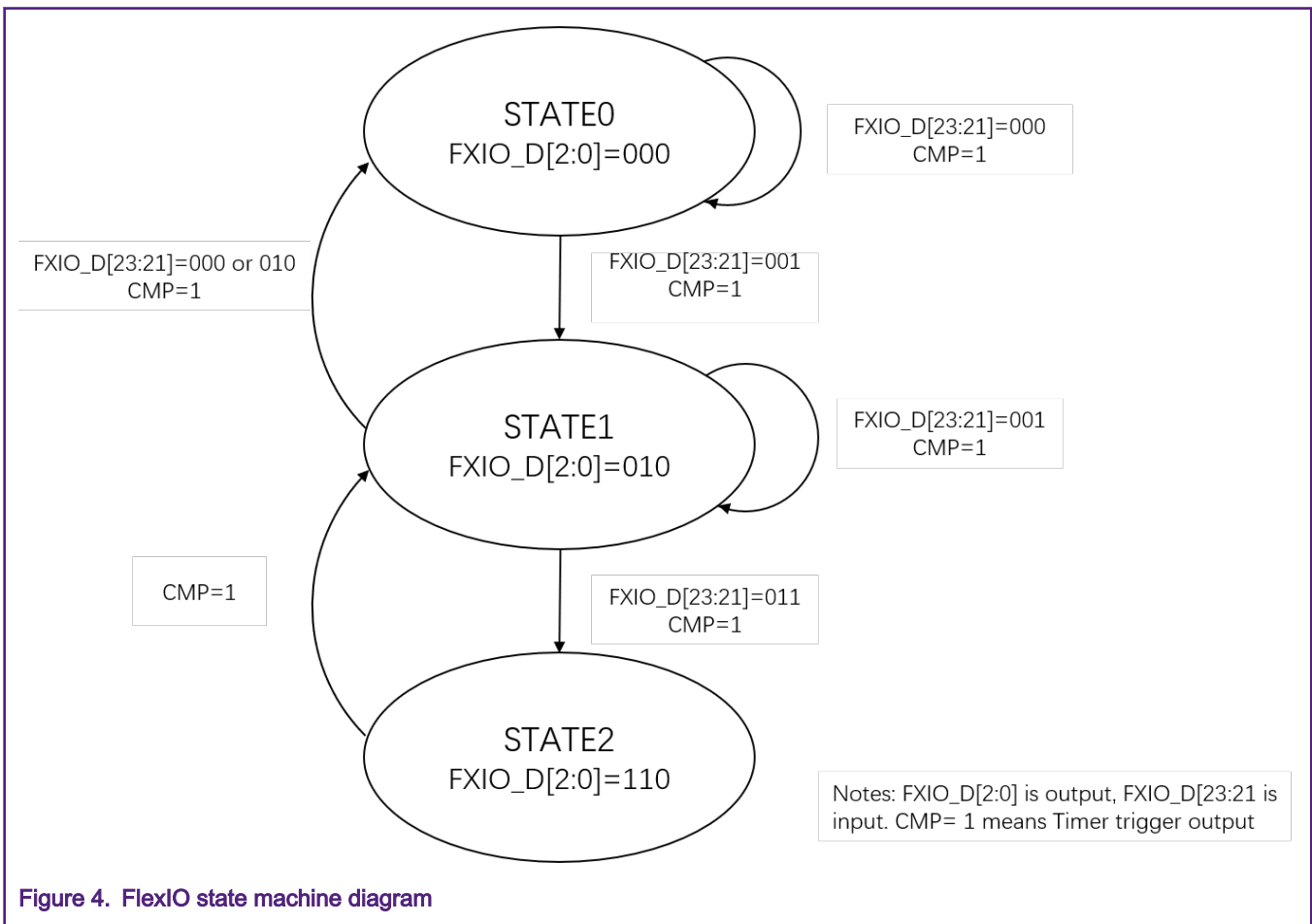


Figure 4. FlexIO state machine diagram

In order to realize the state machine shown in Figure 4, the following registers need to be configured.

Set the PINCFG of FLEXIO01.SHIFTCTL[2:0] to 0x03 and make Shifter pin(FXIO[0], FXIO[1], FXIO[2]) output. Set the PINSEL of FLEXIO01.SHIFTCTL[2:0] to 0x15 and select FXIO[21], FXIO[22], FXIO[23] as state input pins. Set the SMOD of FLEXIO01.SHIFTCTL[2:0] to 0x06 and make Shifter0, Shifter1 and Shifter2 in the state mode.

- FIEXIO01.SHIFTCTL[0] = 0x00831506
- FIEXIO01.SHIFTCTL[1] = 0x00831506
- FIEXIO01.SHIFTCTL[2] = 0x00831506

Set the PWIDTH[3:0] of FLEXIO01.SHIFTCTL [2:0] to 0xF and disable FXIO[7:4]) output. Set the SSTOP[1:0] of FLEXIO01.SHIFTCTL[2:0] to 0x2 and disable FXIO[3] output and enable FXIO[2] output. Set the SSTART[1:0] of FLEXIO01.SHIFTCTL[2:0] to 0x0 and enable FXIO[1:0] output.

- FIEXIO01.SHIFTCFG[0] = 0x000F0020
- FIEXIO01.SHIFTCFG[1] = 0x000F0020
- FIEXIO01.SHIFTCFG[2] = 0x000F0020

In the state mode, the register FLEXIOx.SHIFTSTATE defines the current state. By default (after reset) it is set to 0x0. After a correct start, FLEXIOx.SHIFTSTATE points to the current state, which is defined by the value of register FLEXIOx.SHIFTERBUFn. It's 32-bit value contains the configuration of the current state output (FLEXIOx.SHIFTBUF[31:24]) and the next state selection (FLEXIOx.SHIFTBUF[23:0]). The 24 LSBs value of the FLEXIOx.SHIFTBUF represent eight groups and each group includes three bits. These three bits define the value of the next state, which is selected according to the input combination. For example, if the input combination is 000, the value defined by the FLEXIOx.SHIFTBUF[2:0] bits represents the next state.

- FIEXIO01.SHIFTBUF[0] = 0x00208208
- FIEXIO01.SHIFTBUF[1] = 0x02408408
- FIEXIO01.SHIFTBUF[2] = 0x06249249

Figure 5 shows the value of SHIFTBUF[2:0] and the state jump relationship.

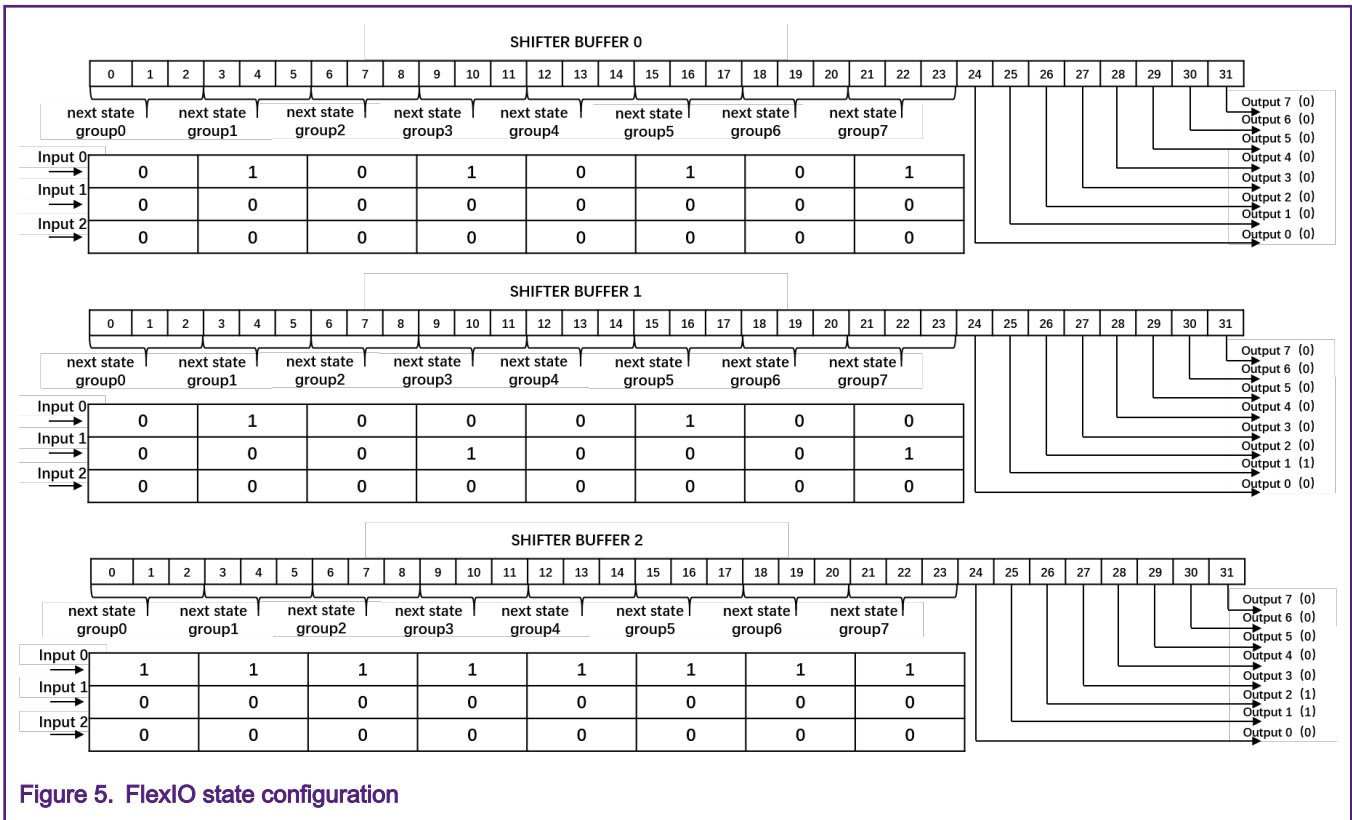


Figure 5. FlexIO state configuration

Shifter0, Shifter1, Shifter2 use Timer0 as the trigger source. Timer0 is configured as 16-bit counter mode, always enable and never reset. The decrement of Timer0 is the FlexIO clock.

- FIEXIO01.TIMCTL[0] = 0x00000003
- FIEXIO01.TIMCMP[0] = 0x0000176F

### 3.3 Low power and clock

FlexIO state machine can be maintained in low power mode. In this case, taking CPU in wait mode as an example, two notes worth paying attention are:

1. When entering low power modes, provide the FlexIO clock remains enabled and the DOZEN is clear, then the FlexIO should remain functional.
2. Set the CG1 of register CCM\_CCGR5 to 0x3 and make sure the flexio1 clock on during all modes except STOP mode.

### 3.4 Running the demo

In this case, when the input value of `FXIO[23:21]` is `011`, FlexIO drives a complementary square wave output at the frequency of `FLEXIO_CLK/12000` on the `FXIO[2]` pin, as shown in Figure 6.

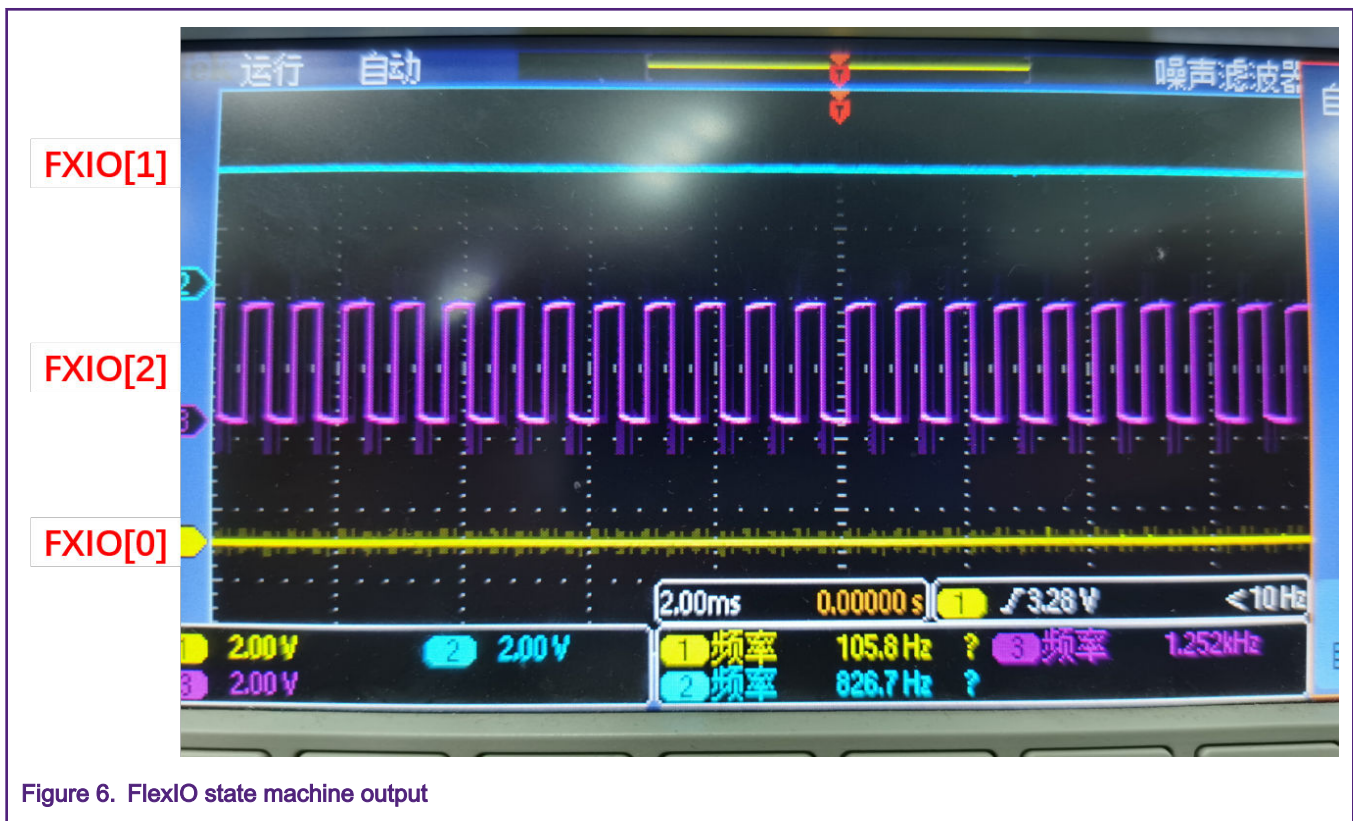


Figure 6. FlexIO state machine output

## 4 Conclusion

This application document introduces the low power state machine function of the FlexIO module and describes the detailed usage method and attention points of FlexIO state mode. Based on its low power feature and flexibility, users can fully offload the CPU and implement specific functions to save hardware resources.

## 5 References

1. *i.MX RT1010 Processor Reference Manual (Rev. B, 07/2019)* (document [IMXRT1010RM](#))

2. *Emulating Hardware State Machine Using FlexIO Module* (document [AN5239](#))

## How To Reach Us

### Home Page:

[nxp.com](http://nxp.com)

### Web Support:

[nxp.com/support](http://nxp.com/support)

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: [nxp.com/SalesTermsandConditions](http://nxp.com/SalesTermsandConditions).

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer's applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, Altivec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, UMEMS, EdgeScale, EdgeLock, eIQ, and Immersive3D are trademarks of NXP B.V. All other product or service names are the property of their respective owners. AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© NXP B.V. 2020.

All rights reserved.

For more information, please visit: <http://www.nxp.com>

For sales office addresses, please send an email to: [salesaddresses@nxp.com](mailto:salesaddresses@nxp.com)

Date of release: May 2020  
Document identifier: AN12845

