Application Note

0380-0243-10 Rev 1.3

Embedded Performance, Inc.
606 Valley Way,   Milpitas, CA 95035
Telephone: (408) 957-0350
FAX: (408) 957-0307
e-mail: support@epitools.com
web: www.epitools.com

MAJIC™ Support For Multi-TAP JTAG Configurations

0380-0243-10 Rev 1.3

November 5, 2003

Introduction

A MAJIC probe performs debug services by using the target processor’s JTAG interface to access registers within its *Debug Support Unit* (DSU). Some target systems provide a dedicated JTAG port for the DSU; connecting a MAJIC probe to such a target is accomplished by simply connecting the JTAG signals as described in the *MAJIC Interface Guidelines* application note for your processor (available on our web site, and your EDT software package installation).

However, most embedded systems use the JTAG interface for much more than just debugging software. Other typical uses include in-circuit programming of flash memory and programmable logic devices, manufacturing test using boundary scan, and access to built in test functions. Furthermore, many embedded systems incorporate multiple processors, and it's desirable to debug the software running on each of them simultaneously through a single JTAG port.

Conceptually, these goals are easily met. The fundamental concept behind JTAG is that the test probe can access each Test Access Port (TAP) on a daisy chain routed throughout the system. However, there are several details which require consideration when creating and using a complex JTAG configuration, especially when building a single device incorporating multiple JTAG TAPs. This application note describes how the MAJIC probe manages complex JTAG configurations, and provides specific recommendations for designing a multi-TAP device.

Note: Chapter 3 of the *MAJIC User’s Manual* presents complete information on the MAJIC Setup Wizard and user defined configuration files, which are mentioned in this application note.

Getting Support

If you have additional questions on this or another topic, please do not hesitate to contact our technical support staff. We are committed to making sure the MAJIC development environment works well for you.

Basic JTAG Connection

Figure 1 shows a simplified view of the data path through a typical JTAG TAP. The MAJIC probe uses the TMS signal to walk the TAP controller through a fairly complex state machine to: select a particular *scan register* via the TDI and TDO multiplexers; capture its data in an N-bit shift register; scan the captured data out on TDO while simultaneously scanning new data in on TDI; then finally update the selected scan register.

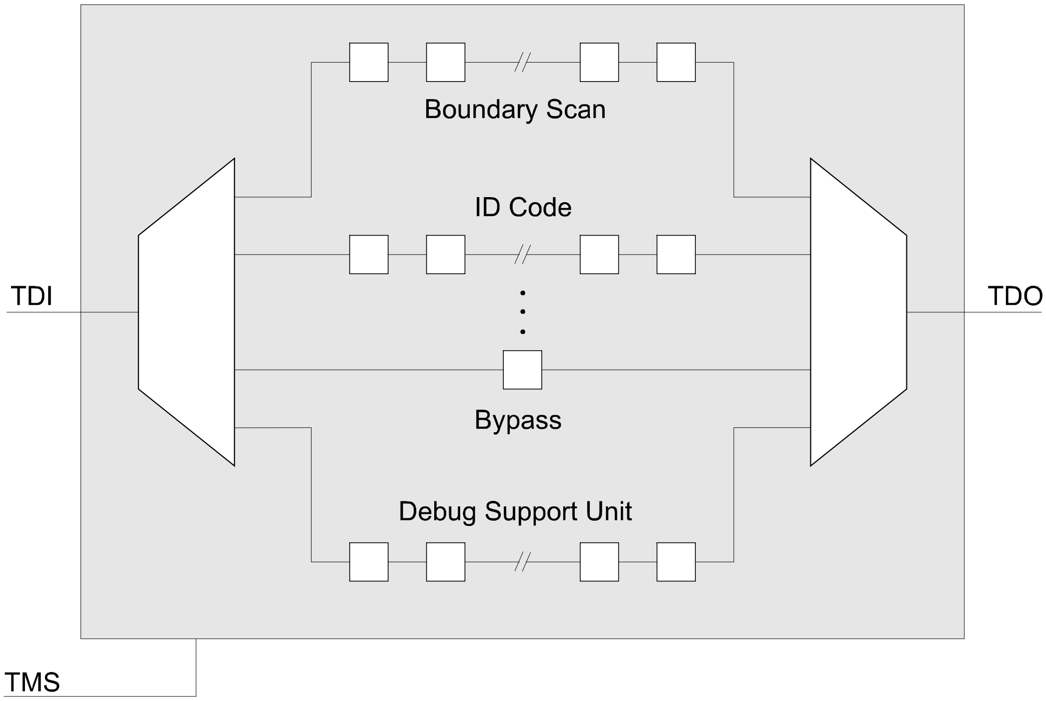

Figure 1: The data path through a single JTAG Test Access Port (TAP). The TAP Instruction Register (not shown) is a special scan register that selects which data scan register is connected between TDI and TDO.

Separate scan registers are used for boundary scan test, reading the device ID code of the part, initiating built-in self-test functions and obtaining their results, and accessing the debug support unit. Although this simplified diagram shows one scan register for the DSU, most DSU implementations actually incorporate several scan registers. These are the registers used by the MAJIC probe to perform debug services. The *TAP Instruction Register* (TAPIR) is a special scan register that controls which of the data scan registers is selected by the TDI and TDO multiplexers. Like the data scan registers, the TAPIR is set by scanning its value in on TDI.

Multi-TAP Configuration

Figure 2 shows a system with four TAPs in a daisy chain. The MAJIC probe still controls the JTAG interface the same way, but the instruction register from each TAP (which selects their scan registers) is concatenated into one long TAPIR register. Likewise, the selected scan register from each TAP is concatenated into one long data register.

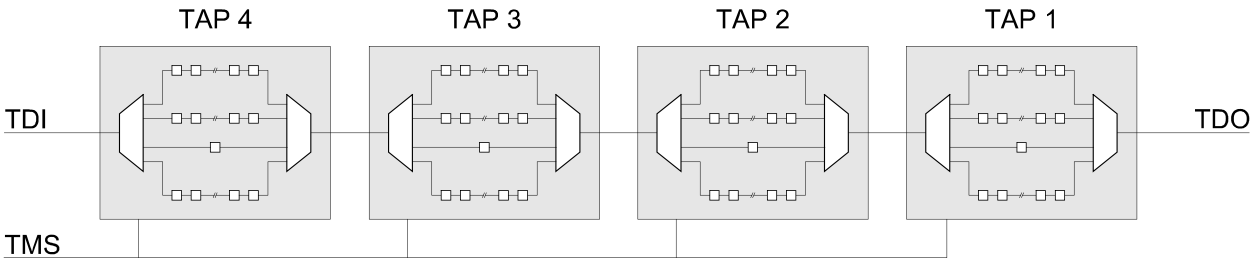

Figure 2: Four daisy chained JTAG TAPs. The TDO from each TAP is chained to TDI of the next to form one long scan chain. TMS is connected to all TAPs so that their state machines operate in unison, but because the instruction registers are concatenated, a different scan register may be selected in each TAP.

When the MAJIC probe wants to communicate with a particular processor, it isolates the corresponding TAP by selecting the bypass register in each of the other TAPs. Then when it scans in the data value, it prepends one extra bit for each bypassed TAP before the processor, and appends one extra bit for each bypassed TAP after the selected processor. For example, to access the DSU of the processor whose TAP is the second in the chain, the MAJIC probe would scan in a BYPASS instruction for TAP 1, then the DSU select instruction for TAP 2, then two more BYPASS instructions for TAPs 3 and 4. After all four instructions have been scanned in, they are all updated in parallel, and all of the TDI/TDO multiplexers in Figure 2 are switched. Then the MAJIC probe scans in one bypass bit for TAP 1, followed by the DSU register value for TAP 2, then two more bypass bits for TAPs 3 and 4. After all the data bits have been scanned through, they are all updated in parallel.

The MAJIC probe handles all of this automatically—the only consideration for the user is that the MAJIC probe needs to know which TAP corresponds to the processor before it can connect to it. The TAP position of the processor is specified with the **`Ice_Jtag_Tap_Select`** option. This option can be set manually after starting the debugger using EDB's Option Settings form (or by entering an **`EO`** command), which is convenient if you want to connect to different TAPs at different times. The **`Ice_Jtag_Tap_Count`** option is a read-only option that reports the number of TAPs in the JTAG scan chain. It can be displayed using EDB's Option Settings window, or with the **`DO IJTC`** command.

If you always want to connect to the same TAP, you can automate the selection with an **`EO`** command in the startup script file. The MAJIC Setup Wizard does not directly support the **`Ice_Jtag_Tap_Select`** option, but you can create a custom board initialization file that includes the following commands, and select that as the user supplied initialization file in the wizard.

```
eo Ice_Power_Sense = _____    // VREF, RST, or TRST, depending on your H/W
eo Ice_Jtag_Tap_Select = ___  // Fill in the processor's TAP position
```

Notes: The TAPs are numbered starting from the one whose TDO signal is connected to the MAJIC probe, as shown in Figure 2. This is because the LSB of TAP 1 is the first bit to be shifted into or out of the scan chain in each JTAG operation.

`Ice_Jtag_Tap_Select` must be set after **`Ice_Power_Sense`**. Since the start-up file created by the wizard sets **`Ice_Power_Sense`** before reading the board initialization file, this detail is only of concern if you create a start up file by hand, or select **`OFF`** for **`Ice_Power_Sense`** in the wizard.

User Defined JTAG Configuration

In most cases the MAJIC probe can automatically determine the JTAG configuration with a quick test when the JTAG interface is first initialized. In some cases, though, it may be necessary to declare the JTAG configuration by writing a JTAG descriptor to the **`MAJIC_JTAG_DIMENSION`** buffer. The format of the JTAG descriptor consists of the number of TAPs in the chain, then a sequence of numbers to specify how many TAPIR bits are in each TAP. When a descriptor is declared in this way, the MAJIC probe will disable the automatic detection process and use the specified descriptor to manage the scan chain.

```
CntTAPs, CntIR1, CntIR2, ..., CntIRn
```

The MAJIC Setup Wizard does not directly support user defined JTAG initialization, because most users never need to worry about it. However, if a user defined descriptor is required, you can create a custom board initialization file that includes the following commands, and select that as the user supplied initialization file in the wizard. It is important to set the JTAG dimension descriptor before the MAJIC probe attempts to initialize the JTAG interface, so you should select **`OFF`** for **`Ice_Power_Sense`** in the MAJIC Setup Wizard, and then set **`Ice_Power_Sense`** as appropriate for your design in your initialization file, as shown below.

```
ew MAJIC_JTAG_DIMSENSION = __,__,__,...,__  // JTAG Configuration Descriptor
eo Ice_Power_Sense = _____      // VREF, RST, or TRST, depending on your H/W
eo Ice_Jtag_Tap_Select = ___    // Fill in the processor’s TAP position
```

Note: In order to figure out the right initialization string, you will need to know the list of devices in your JTAG scan chain, and the order in which they are connected. The EPI technical support group can help you from there.

Multi-Processor Debugging

When debugging multiple processors on a single JTAG scan chain, a separate debug session is started for each processor, and the corresponding TAP is selected within each debug session. The MAJIC probe dynamically changes which TAP is selected and which TAPs are bypassed as it processes debug service requests from each debug session. Within each debug session, the user simply selects the TAP just as when debugging a single processor on a multi-TAP chain; the MAJIC probe handles the rest.

Notes: All models in the MAJIC series support multi-TAP scan chains, but only the MAJIC^{MX} and MAJIC^{PLUS} models support multiple concurrent debug sessions.

The MAJIC^{PLUS} probe only supports a single trace presentation. If your design incorprates multiple processors which have external trace capability, you will need to take additional steps to control which processor's trace interface is presented to the MAJIC^{PLUS} probe. Please contact EPI for additional information on this topic, as it is outside the scope of this application note.

When debugging multi-processor targets, EPI recommends creating a custom board initialization file to define MON command aliases for selecting each processor, and select that file as the user initialization file in the MAJIC Setup Wizard. That way users can select the CPU they wish to connect to by entering a name they can identify instead of having to remember their positions on the JTAG chain. For example, if the command file below is read during initialization, then the user can select the CPU by simply entering its name as a debugger command. Alternatively, you could create separate copies of the start up files for each CPU, and automatically select a different CPU in each.

```
ea CPU_CTL eo Ice_Jtag_Tap_Select = 4  // Select control processor
ea CPU_IO1 eo Ice_Jtag_Tap_Select = 7  // Select I/O processor #1
ea CPU_IO2 eo Ice_Jtag_Tap_Select = 6  // Select I/O processor #2
```

MIPS EJTAG/PCTrace in Multi-TAP Configurations

The EJTAG 1.5 and 2.0 specifications include one aspect that can cause problems in multi-TAP configurations. When PCTrace is enabled, the TDO pin is redefined as a trace signal (TPC). Since this makes it impossible to use the normal JTAG scanning mechanism to halt execution, the TDI signal is redefined as a Debug Interrupt (DINT*) pin. The result is that PCTrace implementations that conform strictly to the EJTAG specification cannot be used if the device is positioned anywhere within a multi-TAP chain.

In order to use EJTAG/PCTrace in a multi-TAP configuration, the EJTAG implementation must support *Demultiplexed PCTrace*. This is an application specific enhancement to EJTAG that keeps the normal mode of operation for TDI and TDO at all times, and uses a separate pin for TPC. Since this means that JTAG scanning may take place while tracing, a separate DINT* pin is not needed in this configuration; instead, the MAJIC probe accesses the DINT bit via a DSU scan register, as it would if tracing were not enabled. The MAJIC^{PLUS} probe supports demultiplexed PCTrace by redefining pin 1 of the JTAG connector via a strap on the MAJIC^{PLUS} Adapter Module (that adapts the MAJIC^{PLUS} to EJTAG); instead of TRST*, pin 1 is used for TPC.

Please contact your CPU vendor to see if their EJTAG implementation supports demultiplexed PCTrace before designing a JTAG scan chain that includes a processor with EJTAG/PCTrace. If not, then you will need to separate each EJTAG TAP onto its own JTAG connector in order to use PCTrace, perhaps with jumpers so that they may be chained during manufacturing test but isolated when debugging software.

Note: It is important to disable the `Ice_Jtag_Use_Trst` option with the MAJIC Setup Wizard if your processor supports Demultiplexed PCTrace.

Multiple TAPs within a Single Device

The MAJIC probe has no way to know where the physical boundaries between devices lie. Fortunately, it does not need to know about the physical partitioning; all it needs to know is how many TAPs there are, and which one corresponds to the processor(s) under test. However, putting more than one TAP within a single physical device may raise issues for other types of JTAG test equipment, because it violates a strict interpretation of the IEEE specification. To be strictly IEEE compliant, each device must have a single TAP, with a single bypass bit.

The MAJIC probe does not rely on this requirement; it assumes that each TAP has one bypass bit, and really doesn’t care how many TAPs there are within each physical device. However, if strict IEEE compliance is important for your application, then a simple daisy chain within your device may not be appropriate. To support sophisticated System-On-Chip designs which use multiple processor and other IP blocks, each with its own TAP controller, EPI recommends the *Nested TAP* architecture shown in Figure 3.

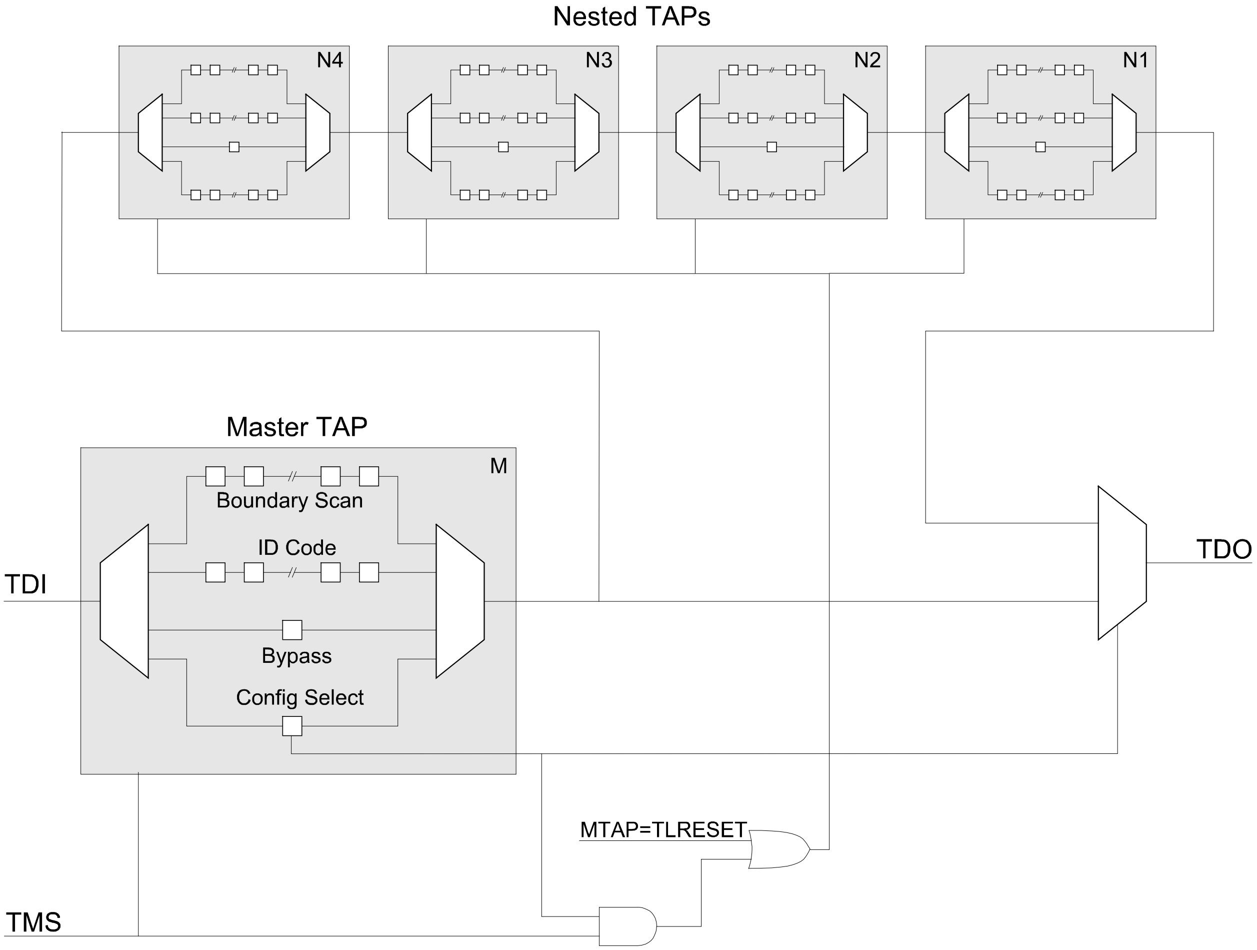

Figure 3: Nested TAP Architecture. The JTAG chain may be expanded to include all five TAPs, or collapsed to include only the master TAP, by setting the Config Select *bit in the master TAP.*

This arrangement includes a *Master TAP* for the device, and a set of *Nested TAPs* which correspond to the various processor cores and other IP blocks within the device. The master TAP is used for boundary scan of the device during manufacturing test. Since the master TAP does not correspond to a processor core, it does not have any DSU scan registers. However, it does provide a special *Configuration Select* scan register for selecting whether or not the nested TAPs are switched into the scan chain.

When the JTAG interface is reset by any means, the Configuration Select bit is cleared, which connects the master TAP's TDO signal to the TDO pin of the device, thereby ensuring that it is the only TAP in the device. As long as the Configuration Select bit remains cleared, the master TAP will be the only TAP, and the device will be IEEE compliant. While operating in IEEE compliant mode, the TMS input to the nested TAPs is forced low to keep them in an idle state, where their TDI input is ignored. Once the Configuration Select bit is set, the TMS input pin is presented to the nested TAPs as well as the master TAP, so that all operate in unison. That way standard JTAG test equipment can be used in manufacturing test environments, but more sophisticated JTAG test equipment such as a MAJIC probe can gain access to the nested TAPs by setting the Configuration Select bit.

Note: In order to ensure that the Configuration Select bit is cleared when the optional TRST* pin is omitted or disabled, the bit should be cleared any time the master TAP state machine is in the TL_RESET state.

Nested TAP Initialization

When using the MAJIC probe, the nested TAPs are mapped into the chain during the JTAG initialization process, and remain enabled for the duration of the debug session. This may be accomplished by writing a scan descriptor to the **MAJIC_JTAG_INIT0** buffer, which defines the JTAG operation that will set the Configuration Select bit. The definition consists of the number of TAPIR bits, followed by the TAPIR bits themselves, then the number of data bits, then finally the actual data bits, as follows:

```
CntIR, TAPIR, CntDR, DR
```

The MAJIC Setup Wizard does not directly support user defined JTAG initialization, because most users never need to worry about it. However, if a user defined descriptor is required, you can create a custom board initialization file that includes the following commands, and select that as the user supplied initialization file in the wizard. It is important to define the JTAG initialization descriptor before the MAJIC probe attempts to initialize the JTAG interface, so you should select **OFF** for **Ice_Power_Sense** in the MAJIC Setup Wizard, and then set **Ice_Power_Sense** as appropriate for your design in your initialization file, as shown below.

```
ew MAJIC_JTAG_INIT0 = __,__,__,__    // JTAG Scan Descriptor
eo Ice_Power_Sense = _____       // VREF, RST, or TRST, depending on your H/W
eo Ice_Jtag_Tap_Select = ___     // Fill in the processor’s TAP position
```

Notes: If there are other devices on the JTAG chain, they must be taken into account as well (examples follow). However, since the nested TAPs are not present until after the initialization operation has been performed, they should not be included in the scan descriptor.

It is not possible (or even advantageous) to dynamically expand and collapse the scan chain. If a JTAG initialization operation has been defined before the MAJIC probe initializes the JTAG interface, then that operation will be performed, and the JTAG chain remain in that state for the duration of the debug session.

One Nested-TAP Device

In this example, assume that Figure 3 is the only device on the JTAG chain, that the master TAP has a 6-bit instruction register, and that the TAPIR value for selecting the Configuration Select scan register is 101100. The following JTAG initialization command defines a JTAG initialization operation having six TAPIR bits, whose value is 101100, and one data bit whose value is 1:

```
ew MAJIC_JTAG_INIT0 = 6, 2C, 1, 1
```

After this operation, the JTAG chain includes all five TAPs as shown below. If the processor is the fourth nested TAP (N4), then the **Ice_Jtag_Tap_Select** option should be set to **4** to select the processor.

—TAP^{M}—TAP^{N4}—TAP^{N3}—TAP^{N2}—TAP^{N1}—

One Nested-TAP Device Between Other Devices

In this example, assume that the Nested-TAP device in the first example is actually TAP 3 in Figure 2. Furthermore, assume that TAPs 1 and 2 have three TAPIR bits each, and that TAP 4 has five TAPIR bits. The following JTAG initialization command defines a JTAG initialization operation having seventeen TAPIR bits, whose value is 11111-101100-111-111 (the underlined instruction bits are the BYPASS commands for TAPs 1, 2, and 4) and four data bits 1111. (the underlined data bits are the BYPASS scan register bits for TAPs 1, 2, and 4).

```
ew MAJIC_JTAG_INIT0 = 0n17, 1FB3F, 4, F
```

Note: The BYPASS command is defined as all ones, no matter how may bits are in the given TAPIR register. The bypass data value is irrelevant.

After this operation, the JTAG chain would have eight TAPs as shown below. If the processor is the fourth nested TAP (N4), then the `Ice_Jtag_Tap_Select` option should be set to **6** to select the processor.

—TAP4—TAP3^{M}—TAP3^{N4}—TAP3^{N3}—TAP3^{N2}—TAP3^{N1}—TAP2—TAP1—

Four Nested-TAP Devices

In this example, assume that Figure 2 depicts four instances of the Nested-TAP device in the first example, and that you want to expand TAPs 2 and 3 but not 1 or 4. The following command defines a JTAG initialization operation having 24 TAPIR bits, whose value is 111111-101100-101100-111111 (the underlined instruction bits are the BYPASS commands for TAPs 1 and 4) and four data bits 1111. (the underlined data bits are the BYPASS scan register bits for TAPs 1 and 4).

```
ew MAJIC_JTAG_INIT0 = 0n24, FECB3F, 4, F
```

After this operation, the JTAG chain would have twelve TAPs as shown below. To connect a debug session to the processor in the second device (TAP2^{N4}), `Ice_Jtag_Tap_Select` option should be set to **5** in that debug session. To connect another debug session to the processor in the third device (TAP3^{N4}), the `Ice_Jtag_Tap_Select` option should be set to **10** in that debug session.

—TAP4^{M}—TAP3^{M}—TAP3^{N4}—TAP3^{N3}—TAP3^{N2}—TAP3^{N1}—
—TAP2^{M}—TAP2^{N4}—TAP2^{N3}—TAP2^{N2}—TAP2^{N1}—TAP1^{M}—