



Designing with JTAG In-System Programmable AMD MACH CPLDs

Application Note

Designing with JTAG In-System Programming AMD MACH CPLDs



Application Note

by Arthur Khu

The AMD MACH[®] JTAG CPLD devices are high-density, high-speed EE CMOS programmable logic devices with predictable timing delays and IEEE 1149.1 (JTAG) test access port (TAP) pins for board testability. These devices can be programmed in-system through the TAP pins. MACH CPLD users can follow some design techniques so that after a design has been fitted and a board layout made, they can take advantage of this on-board programming capability to modify or add logic to the design while maintaining the timing and existing pinouts. These techniques are called balanced partitioning and spread placement.

ADVANTAGES OF JTAG IN-SYSTEM /ON-BOARD PROGRAMMING

MACH CPLDs with JTAG in-system programming (JTAG-ISP) capability offers numerous design advantages in prototyping, manufacturing, and field-upgradability. AMD supplies a PC-based software tool called MACHPRO[®], to program MACH JTAG-ISP devices on a board through a programming cable connected to the PC parallel port. Logic changes to MACHs on prototype systems can be made quickly by programming the MACHs on-board through the TAP pins. MACH JTAG CPLDs can be included in a serial JTAG chain with other non-MACH JTAG-compliant devices. MACHPRO will put the other non-MACH devices into bypass mode when configuring the MACHs.

MACHPRO can also generate vector files that are compiled as test programs on Automated Test Equipment (ATE). This enables boards containing MACH JTAG-ISP CPLDs to be programmed and tested on the same board tester. Once the boards are out in the field, they can still be upgraded via the programming cable if the customer or field engineer has access to the TAP pins. Another option is to send the updated MACH programming pattern to customers on disk. The configuration data will include programming instructions in the form of test vectors specifying the state of each TAP pin. If your system has a microcontroller, you can develop a routine to download the test vectors on the update disk and then apply each vector to the JTAG TAP pins. Since the configuration data is embedded with the programming instructions, your programming routine can configure MACHs in any JTAG chain with other non-MACH JTAG devices.

JTAG SIGNAL INTEGRITY

The JTAG TAP consists of 4 pins: Test Data Input (TDI), Test Data Output (TDO), Test Clock (TCK), and Test Mode Select (TMS). TMS and TCK are connected in

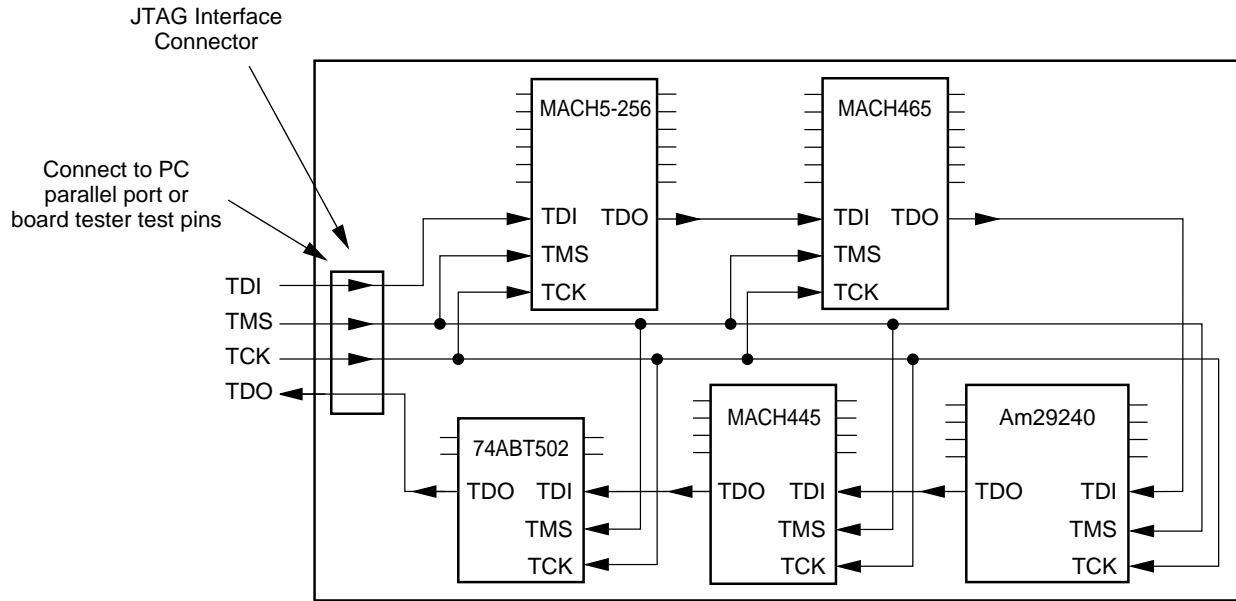
parallel to all the JTAG devices in a chain, while TDI and TDO are connected in series (Figure 1).

The MACH355, MACH445, and MACH465 have the optional JTAG Test Reset TRST(L) pin and an optional programming ENABLE(L) pin. The TRST pin is used to asynchronously reset the JTAG test circuitry, and the ENABLE pin is a hardware program enable signal. Both of these input pins have internal pull-ups.

If you want to take advantage of the asynchronous test reset capability, you can connect TRST on your board to the TRST line on the MACHPRO programming cable. If you want to work with the 4-pin JTAG interface (i.e., TCK, TMS, TDI, and TDO), you can leave the TRST line unconnected (because there is an internal pull-up) or you can externally pull it high. MACHPRO can still synchronously reset the JTAG circuitry by holding TMS high for at least five clock cycles. MACHPRO does a synchronous test reset at the beginning and end of any JTAG operation it performs.

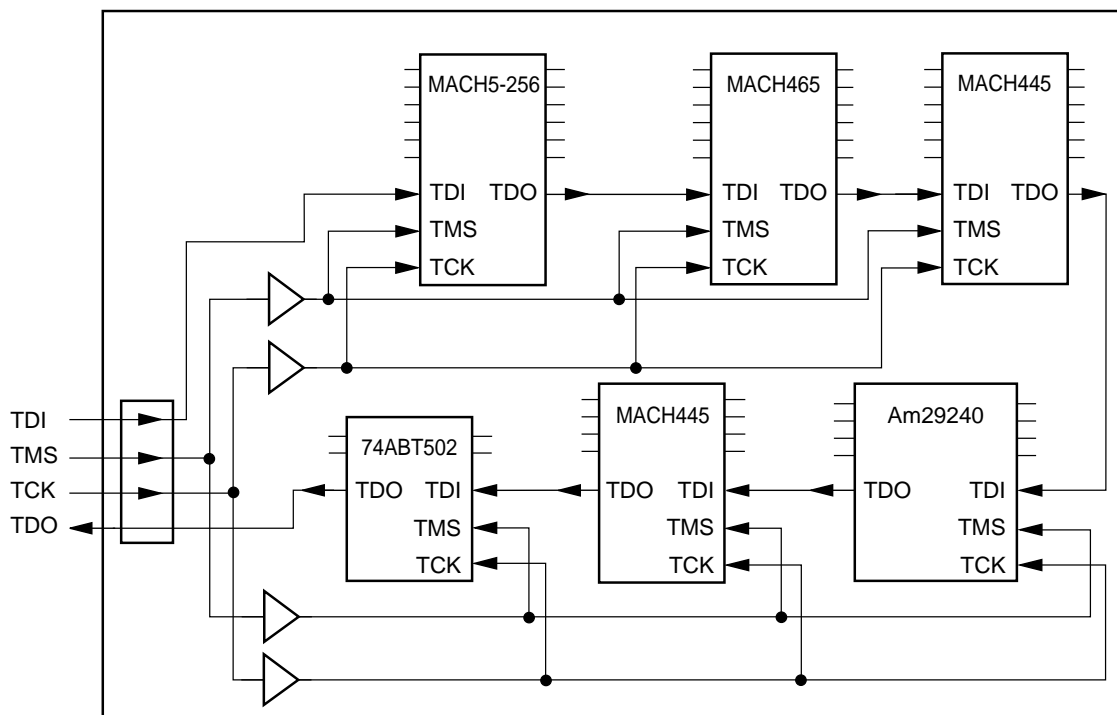
For MACH JTAG CPLDs with ENABLE lines, the ENABLE line has to be LOW to enable programming. Once the part is program enabled, programming then occurs only when the exact sequence of programming passwords, JTAG instructions, and data are applied in the correct order through the JTAG TAP pins. You therefore cannot accidentally program the part even if ENABLE(L) is LOW. You can connect the ENABLE input to the programming cable to have the MACHPRO software drive this line, or you can connect the ENABLE(L) line LOW to always enable the part.

When designing the board, minimize the trace lengths of the TCK and TMS lines. If the number of JTAG devices in a chain is greater than 6, insert a buffer (e.g., 74367) to ensure the integrity of the TCK and TMS signals. You may also put multiple buffers to balance the distribution of the TCK and TMS signals if the number of JTAG devices in a chain is large and the traces are long (Figure 2).



21149A-1

Figure 1. Board with 5 JTAG devices connected in a serial JTAG chain. TDI and TDO are connected in series while TMS and TCK are in parallel.



21149A-2

Figure 2. Use buffers to drive and balance the distribution of TMS and TCK lines on a board with six or more JTAG devices connected in a serial chain, or if the JTAG devices are physically placed far apart.

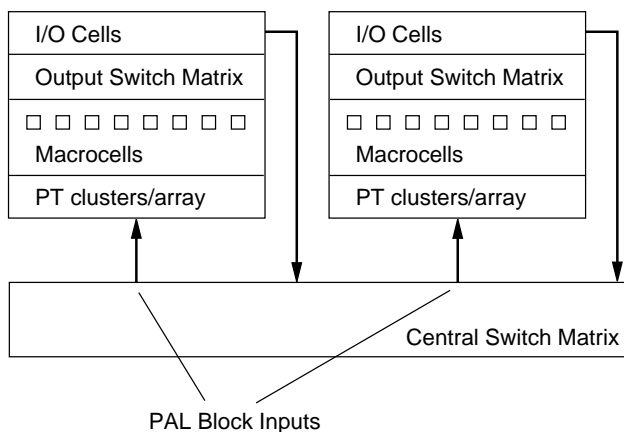
DESIGN PLANNING

MACH JTAG-ISP devices can be programmed in-system, but this also requires that the new logic configuration use the existing MACH pin outs. By using the balanced partitioning and spread placement design techniques, you can take advantage of the in-system configurability of MACH JTAG CPLDs to implement

design changes while maintaining the MACH pin placements on the board.

BALANCED PARTITIONING

The current generation of MACH CPLDs has multiple PAL-like blocks interconnected through a routing matrix. Each PAL block has a certain number of logic macrocells and block inputs available (Figure 3).



21149A-3

Figure 3. PAL blocks are interconnected by a routing matrix.

Fitting logic equations into a device means assigning an equation to one of the macrocells in the blocks and routing the signals required by this equation into the block. With the balanced partitioning technique, equations are placed into blocks such that the number of inputs going into each block is balanced or equalized.

For example: A design has the following logic equations

$$X = A*B + C*D$$

$$Y = E*F + G*H$$

$$Z = A*/B + K*L$$

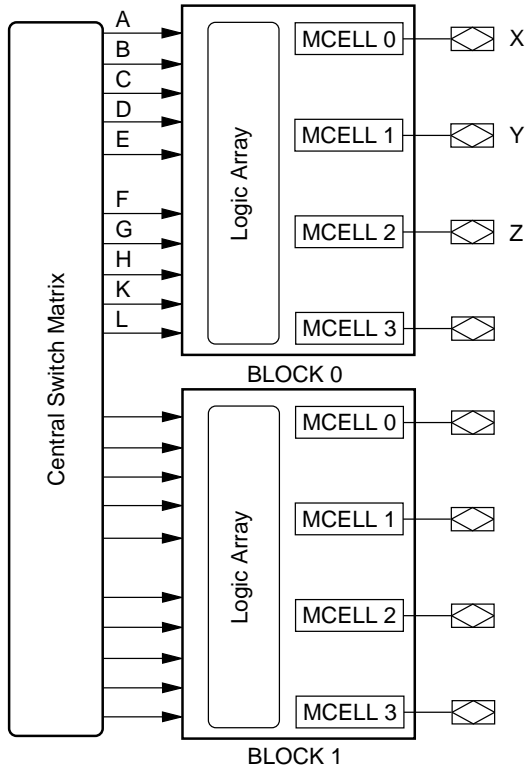
Assume a device has 2 blocks labeled block 0 and 1 and each block is composed of 4 macrocells with 10 block inputs each. If equation X is placed in block 0, then X will use one macrocell in block 0, and will require 4 block array inputs to route the variables A, B, C, and D. If Y is also put into a macrocell in block 0, then Y will require another 4 block inputs to route the variables E, F, G, and H. Placing the Z equation in block 0 will use

only 2 additional block inputs for the variables K and L because A and B were already routed into block 0 for the X equation.

The total number of block 0 inputs used is 10 (see Figure 4a).

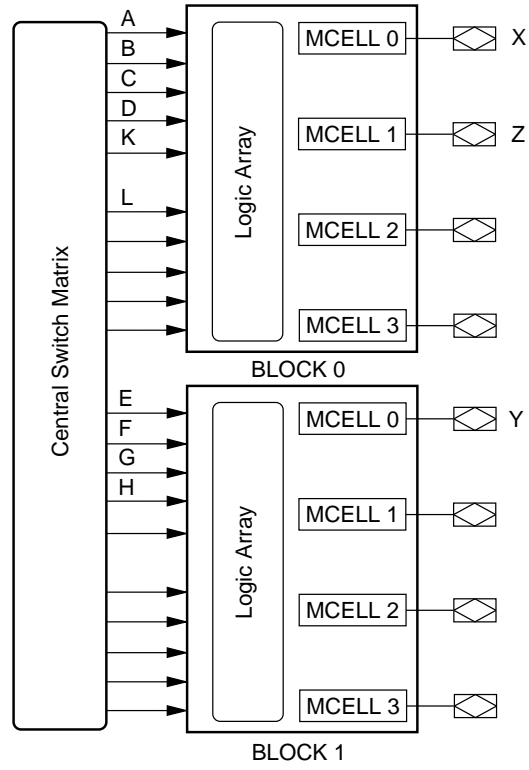
Since equations X, Y, and Z were all placed into block 0, the 4th macrocell in block 0 can only be used to implement an equation which is a function of the 10 variables already using the block inputs. The designer cannot modify the existing logic equations in block 0 to include more variables because there are no more block inputs available.

Balanced partitioning will spread the logic equations across blocks so that there will be block inputs available for logic additions or modifications. In Figure 4b, a recommended signal assignment is to put equations X and Z in the same block since they share some variables, and Y into the other. This logic partition requires only 6 block array inputs instead of 8 if signals X and Y were put into block 0.



21149A-4

Figure 4a. Signals X, Y, and Z are placed in the same PAL block.



21149A-5

Figure 4b. Signals X, Y, and Z are spread across multiple blocks.

MACH CPLD design software usually performs this automatically, but sometimes the fitter partitioning software may not group the right set of signals into the same block. To override this, MACH design software such as AMD's MACHXL and Data I/O's ABEL provide a "grouping" capability so that the designer can instruct the fitter to put certain signals together.

MACHXL uses the "GROUP MACH_SEG_x" statement where x is the block letter:

```
PIN ? x
PIN ? y
PIN ? z

GROUP MACH_SEG_A x z;
GROUP MACH_SEG_B y;
```

In ABEL, the GROUP property is used:

```
DECLARATIONS
x, y, z PIN;

AMDMACH property 'GROUP A x z';
AMDMACH property 'GROUP B y';
```

The grouping capability can also be used to put or pack as many signals as possible into the fewest number of blocks so that the remaining blocks in a CPLD are left unused and reserved for future logic additions to the device.

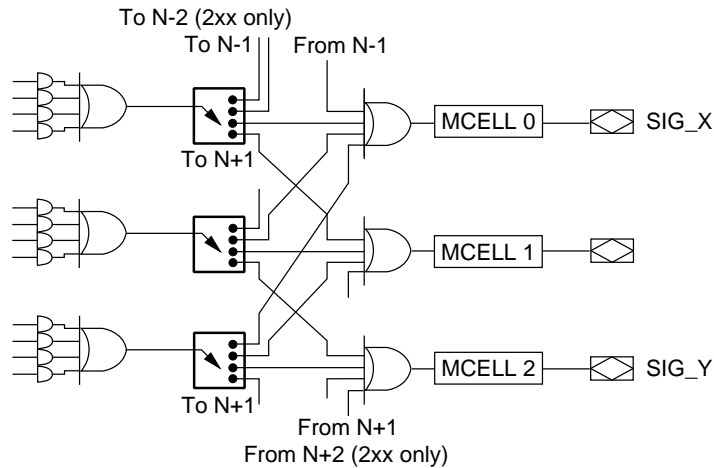
SPREAD PLACEMENT

Spread placement refers to the spreading or spacing apart of equations within a block so that there will be unused PT clusters on either side of a signal (Figure 5). In the event that more product terms are added to a signal, the unused clusters on either side of the modified equation can be steered to it.

In figure 5, SIG_X is assigned to the IO pin attached to macrocell 0, and SIG_Y is assigned to macrocell 2. If logic changes are made to SIG_X that increase the number of product terms required, then it can steer the product terms away from macrocell 1 which is currently unused. If SIG_Y had been assigned to macrocell 1, then it is still possible to steer macrocell 1's product terms to macrocell 0, but now this means that SIG_Y will have to obtain its product terms from macrocell 2 and beyond.

This technique can be implemented by assigning equations to every other IO pin, or by reserving pins for future use. In ABEL, pins are reserved by assigning a signal name to a pin but not using the pin in any equation. In MINC and MACHXL, designers can reserve pins and product terms by assigning temporary pin names and fabricated logic equations to the pins. When the pins and product terms are required for design changes, additional resources can be released by removing the temporary pins and logic equations.

Note that the timing is fixed regardless of which macrocell the steerable product term cluster was steered from. In the MACH1xx device family, a macrocell can use its product term group, the product term group from the macrocell above it and the one from below it. In the MACH2xx, MACH3xx, and MACH4xx family, a macrocell can also steer product terms from a macrocell 2 steps away from it. In figure 5, macrocell 0 can steer and use the product terms from macrocell 2.



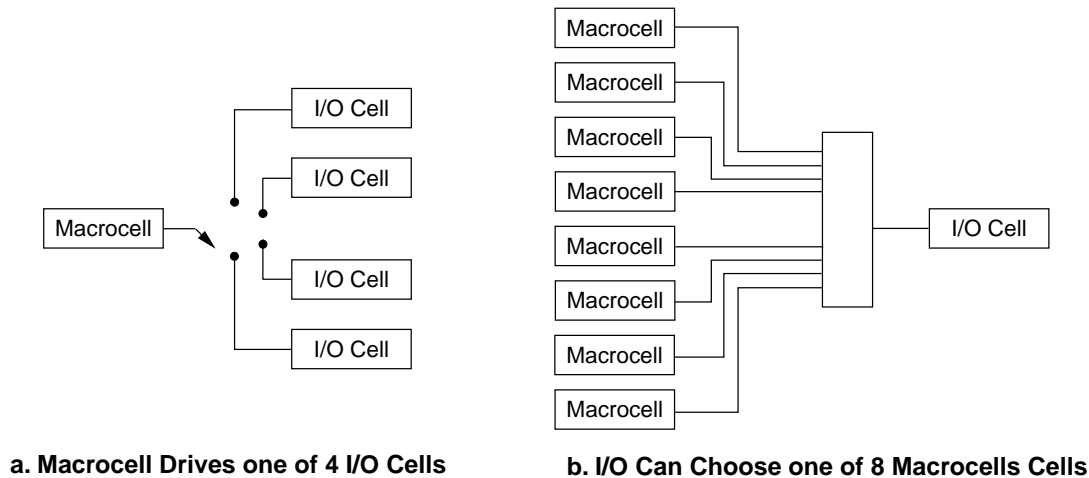
21149A-6

Figure 5. Spread placement in a MACH2xx device. Macrocell 1 PTs can be steered to mcell 0 or 2.

OUTPUT SWITCH MATRICES

The MACH3xx/4xx device family has many additional architecture features over the MACH1xx/2xx. In addition to bigger product term groups (5 product terms instead of 4), input registers, and synchronous/asynchronous capability for the macrocells, there are also output switch matrices in the MACH3xx/4xx.

These output matrices are multiplexers between the logic macrocells and IO pins which allow macrocells to go to different IO pins (Figure 6a). Conversely, IO pins can reach different macrocells (Figure 6b). This feature is helpful in handling design changes while maintaining preassigned pinouts. Since there is no speed penalty for using the output matrices, propagation delays remain constant throughout the part.



21149A-7

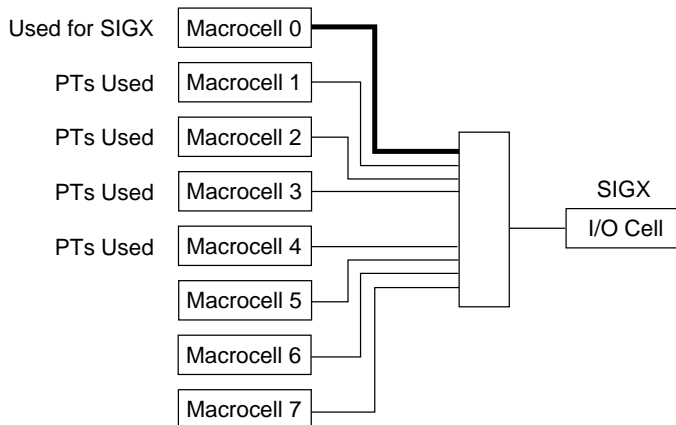
Figure 6. MACH4xx Output switch matrix.

A designer uses CPLD software such as MACHXL or ABEL to process a design and obtain a pinout. If a signal is not preassigned to a pin in the design file, then the pinout is determined by the software. If the signal is fixed or preassigned to a pin, then the PT steering arrangement is determined by the pin assignment. When the user changes the design by adding another product term to an equation, it is possible that the current PT cluster steering scheme can become invalid.

For example, if the number of PTs in output signal SIGX in figure 7a is increased from 5 to 6, then SIGX now requires another PT cluster. Since all the PT clusters available to SIGX have already been steered away

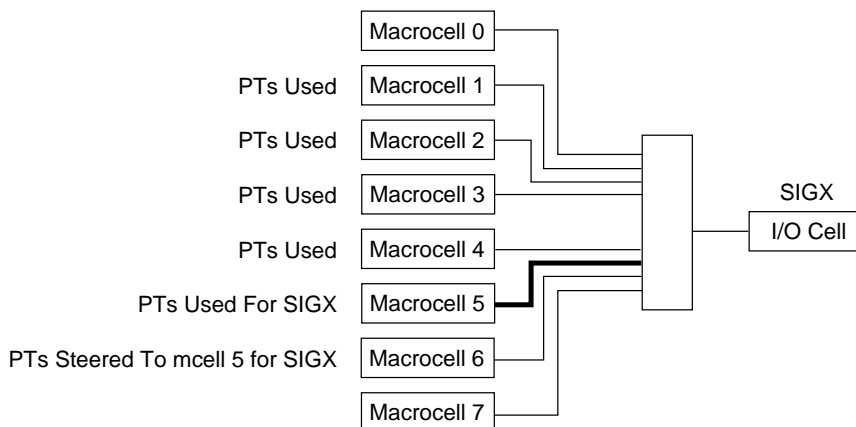
to other macrocells, this macrocell placement is not valid. If the CPLD did not have an output matrix, then the pinout must change to accommodate a new PT steering arrangement. Since the MACH3xx/4xx parts have output matrices, SIGX can be moved to another macrocell which can still reach the previously assigned pin (Figure 7b).

To maintain the pinout, most MINC or ABEL MACH design source files do not need to be changed because the CPLD tools that support output switch matrices will automatically reassign SIGX to a new macrocell location that can still reach the assigned IO pin.



21149A-8

Figure 7a. Macrocell 0 is used for SIGX.



21149A-9

Figure 7b. PTs from Macrocell 5 and 6 are used by SIGX which is assigned to Mcell 5.

Spread placement as a planning technique for maintaining pinouts while allowing for future design changes is not required in MACH3xx/4xx devices because these parts have the output switch matrices. If an equation is modified that increases the number of PTs required by the equation, the software automatically searches for a macrocell capable of supporting the increased PT requirements of the modified equation while retaining the existing pin assignments.

The spread placement technique works in conjunction with the balanced partitioning scheme for all other AMD MACH products. Signals are spread across blocks to reduce the number of macrocells and block inputs used in each block, and the equations in a block are spread across the macrocells to allow room for logic expansion.

Trademarks

Copyright © 1996 Advanced Micro Devices, Inc. All rights reserved.

AMD, the AMD logo, and combinations thereof are trademarks of Advanced Micro Devices, Inc.

MACH and MACHPRO are registered trademarks of Advanced Micro Devices, Inc.

Product names used in this publication are for identification purposes only and may be trademarks of their respective companies.