

ARM Cortex magú mikrovezérlők

Debug-olás

Scherer Balázs



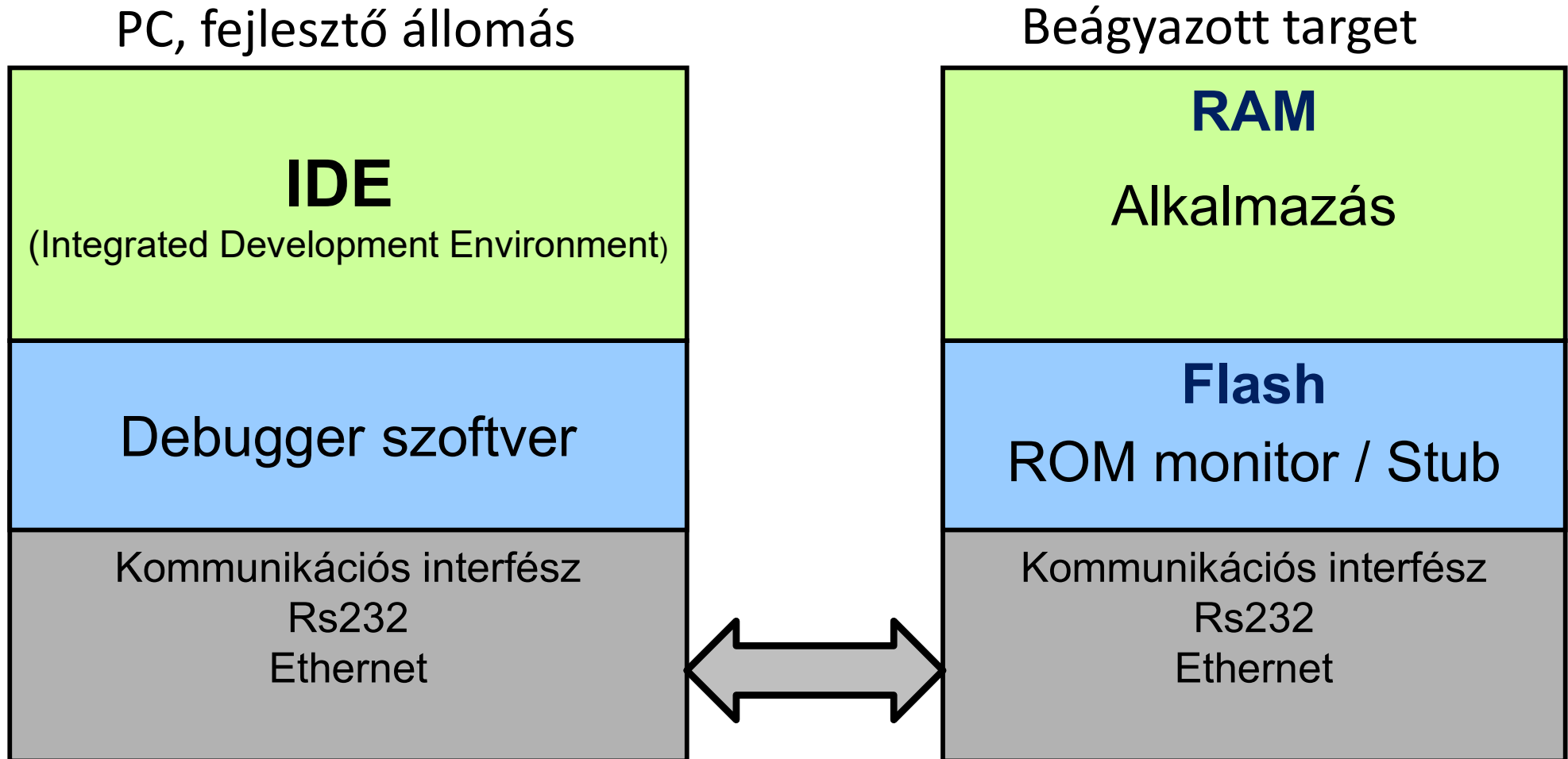
Méréstechnika és
Információs Rendszerek
Tanszék

Tradicionális debug módszerek

- Hagyományos debug
 - Oscilloscope
 - Logikai analizátor
 - Printf
 - LED villogtatás
- In-Circuit Emulatorok
 - Debug variáns
 - Dual port RAM
 - Hardware-es break pointok
 - Regiszter figyelés

Debug módszerek 2000-ig

- ROM monitor: GDB stub



ROM monitor

Rom monitor

- Csak a RAM terület használható az alkalmazás által
 - Régi külső RAM-os 32 bites megoldások
- Kommunikáció a ROM monitor és a debugger között
 - GDB: Remote Serial Protocol

Debugger alkalmazás



Méréstechnika és
Információs Rendszerek
Tanszék

GDB: GNU Debugger I.

- GDB alapvetően egy parancssori debugger, amihez vannak megjelenítők
 - DDD
 - Eclipse

Minta GDB session:

```
localhost$ gdb a.out // A GDB elindítása
GNU gdb 5.0
Copyright 2000 Free Software Foundation, Inc.
(gdb) set remotebaud 57600
(gdb) target rdi com1 // connect to target machine, process, or file
(gdb) load // dynamically link file and add its symbols
Loading section .text, size 0x1280 vma 0x1000
Loading section .data, size 0x760 vma 0x2280
Loading section .stack, size 0x10 vma 0x30000
Start address 0x1000
Transfer rate: 53120 bits in 1 sec.
```

GDB: GNU Debugger II.

```
(gdb) break main           // breakpoint b [file:]line or function
Breakpoint 1 at 0x8048476: file test.c, line 5.
(gdb) continue           // continue running your program
Breakpoint 1, main () at test.c:5
5 for( i = 0; i < 10; i++ ) {
(gdb) display j           // show value of expr each time program stops [according to
format f ]
1: j = 1074136126
(gdb) step               // stepping program
6 j = i * 2 + 1;
1: j = 1074136126
(gdb) step               // stepping program
5 for( i = 0; i < 10; i++ ) {
1: j = 1
(gdb) quit              // quitting debugg session
```


GDB: Parancsok

- ***run***: Program futtatás
- ***continue***: program folytatás
- ***next***: következő utasítás
- ***step***: következő utasítás step-into
- ***list***: forrás kód ki listázás
- ***break***: break point elhelyezés (hw-es breakpointok néha macerások)
 - ***disable/enable***
- ***print***: változó értékének kiírása
- ***set***: változó érték adása

És még sok egyéb

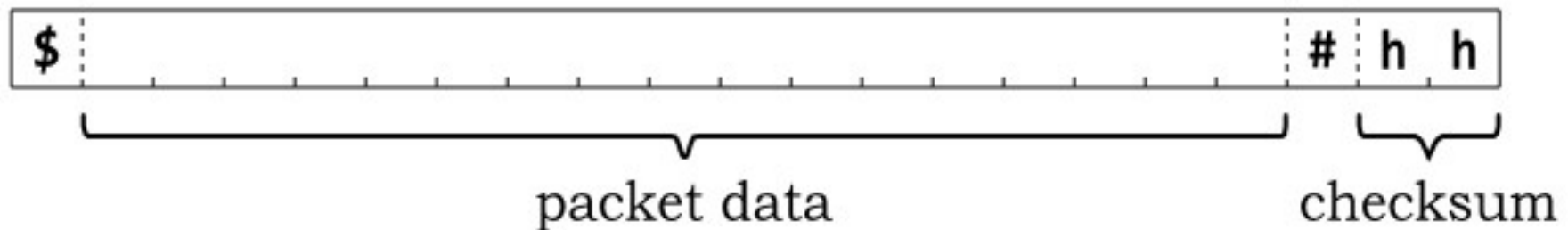
Debugger, ROM monitor kapcsolat



Méréstechnika és
Információs Rendszerek
Tanszék

GDB: Remote serial protocol

- Valahogy tartani kell a kapcsolatot a target-tel
 - Rs232
 - TCP protokoll
- ASCII karakterekből álló parancsok
 - \$-jellel kezdődnek
 - #-al majd egy 8bites cheksum-al záródnak
- A stub
 - + al válaszol az elfogadott – al a hibás checksumra.
 - „OK”-val, vagy hibakóddal a parancsra



Gyakran használt RSP parancsok

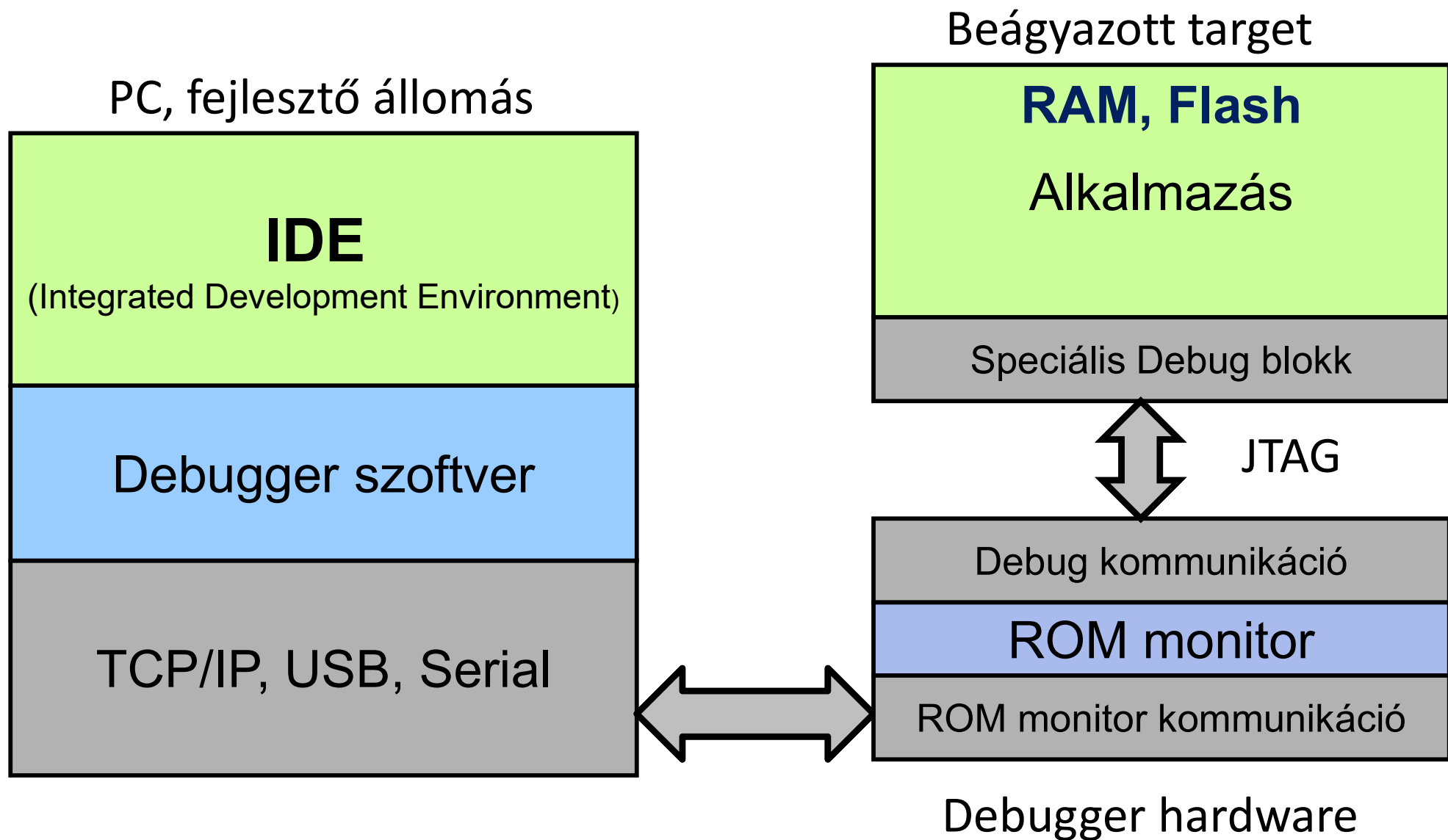
- **Read Registers (g):** kiolvassa a target összes regiszterét:
- **Write Register n (P):** egy specifikus regiszter értékét írja
- **Read Memory (m):** egy memória területet olvas
- **Write Memory (M):** egy memória területre ír.
- **step (s):** Lépteti egyet a processzort. A + on kívül nincs rá más válasz.
- **Continue (c):** A rendszer folytassa a program végrehajtást normál sebességgel
- **Breakpoint-ok (Z0 packet):**
 - A GDB alpból szoftver breakpointokat igyekszik rakni (**Z0** packet)
 - TRAP utasítással felülírja az adott memória címet.
 - **Z1** hardware-es breakpoint (limitált).

Letöltött monitor nélküli debuggolás 2000 után

- Emulátor tulajdonság szükséges
 - Változók értékék megfigyelése
 - Változók értékék megváltoztatás
 - Step-elés
 - Breakpoint állítás
- JTAG-en keresztüli debuggolás
 - JTAG alapvetően csak egy kommunikációs séma
 - 5 vezetékes
 - TDI, TDO, TMS, TCK, nTRST

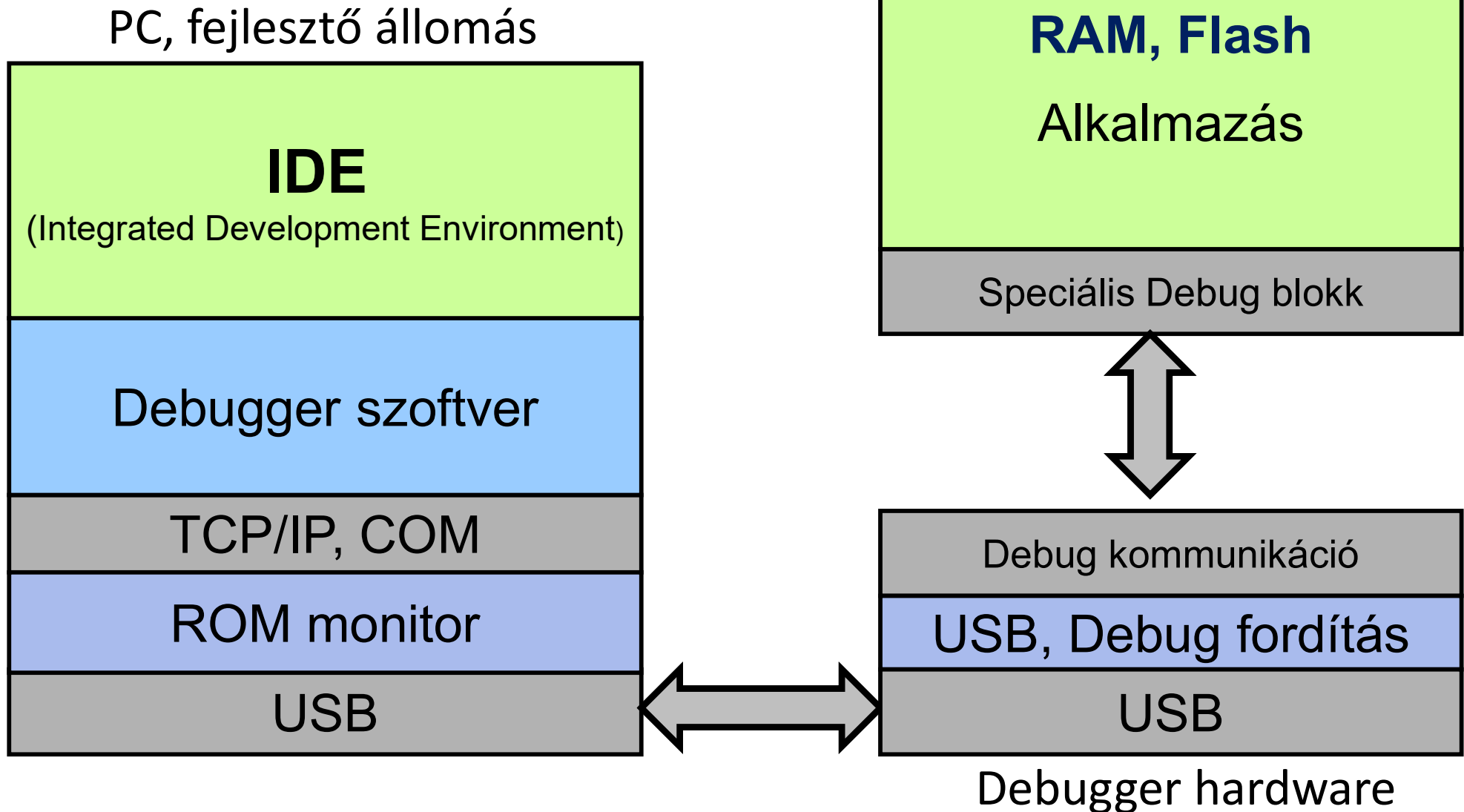
JTAG alapú debuggolás

- ROM monitor átalakul



JTAG alapú debuggolás

- ROM monitor átalakul, sokszor bekerül a PC-re
Beágyazott target



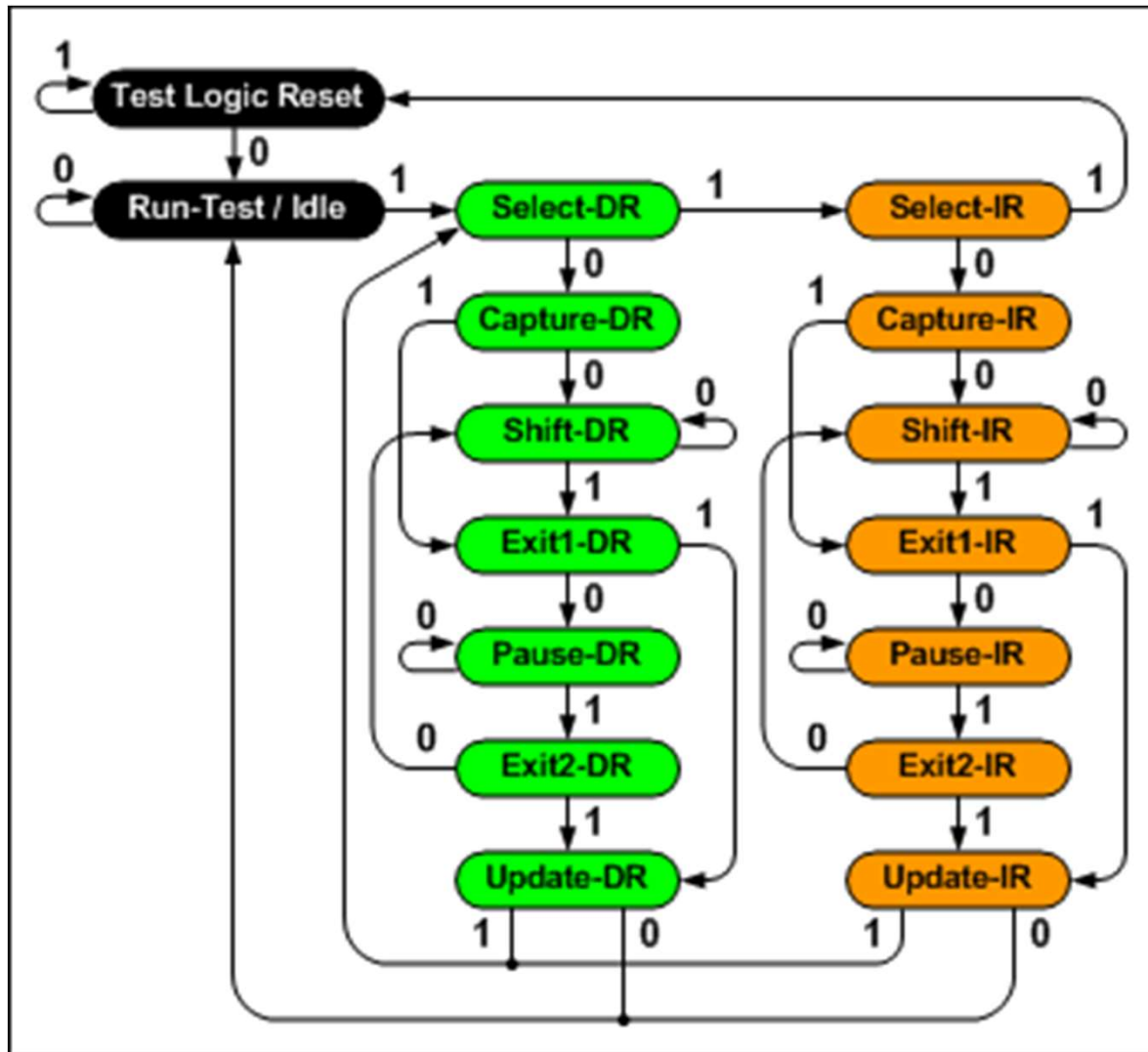
Debugger kommunikáció/hardware



Méréstechnika és
Információs Rendszerek
Tanszék

JTAG (Joint Test Access Group)

- Állapotgép



JTAG eszközök

■ FT2232(H)

- USB alapú JTAG megoldás
- Az összes újabb „olcsó” JTAG ezt használja
- 30 MHz-es JTAG clock



20-PIN JTAG/SW Interface

VCC	1	□	□	2	VCC (optional)
TRST	3	□	□	4	GND
TDI	5	□	□	6	GND
SWDIO / TMS	7	□	□	8	GND
SWCLK / TCLK	9	□	□	10	GND
RTCK	11	□	□	12	GND
SWO / TDO	13	□	□	14	GND
RESET	15	□	□	16	GND
N/C	17	□	□	18	GND
N/C	19	□	□	20	GND

Integrált Debug monitor



Méréstechnika és
Információs Rendszerek
Tanszék

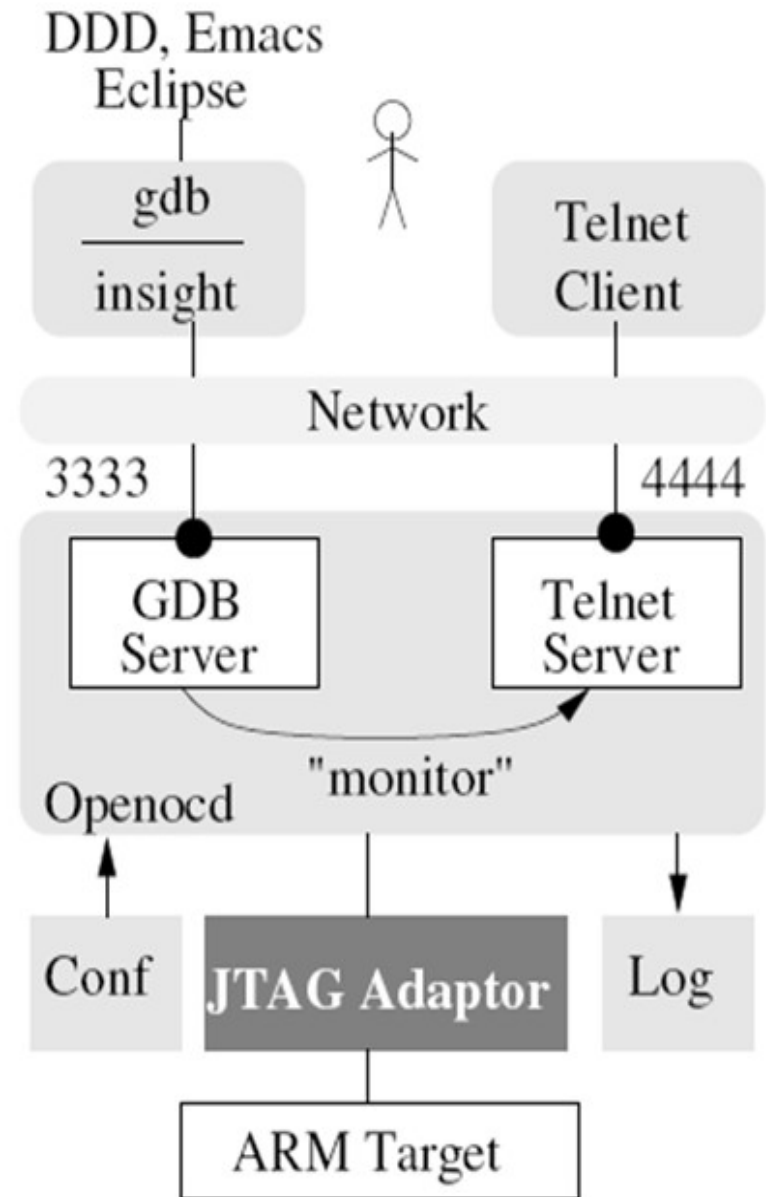
JTAG, GDB csatolás

- ***A debugger szerver és debugger közötti USB kommunikáció sokszor gyártó függő és nem nyilvános***
 - *ST-Link*
 - *J-Link*
- ***Sokszor a debugg szerver is el van rejtve és az is gyártó függő***

JTAG, GDB csatolás

■ *Open OCD*

- GDB server
- JTAG támogatással
- RSP parancsok konverziója JTAG parancsokká
- Külön konfigurációs port
 - Server handling
 - Target management
 - JTAG memory management
 - Flash memory management



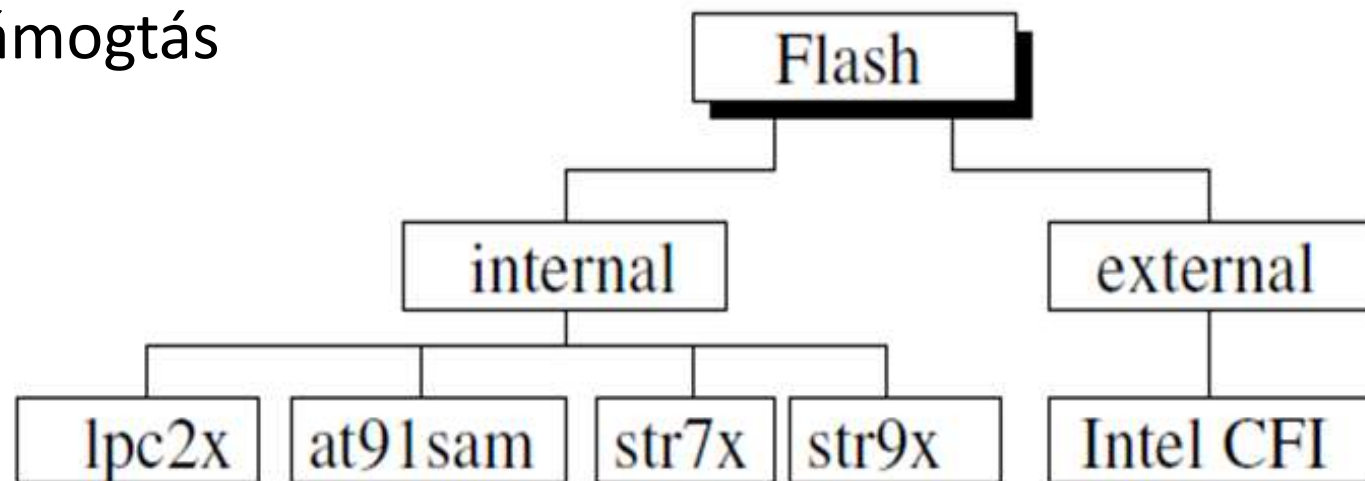
Az OpenOCD fordítás függő részei

■ Target modul

- Minden target core variánshoz külön c file

■ Flash modul

- Hardware függő Flash támogatás



Az OpenOCD konfiguráció

- **Konfigurációs rész minden target-hez**
 - Server (daemon) konfiguráció
 - JTAG konfiguráció
 - JTAG scan chain konfiguráció
 - Target konfiguráció
 - Flash konfiguráció

Példa egy STM32 konfiguráció-ra

Parancs:

```
openocd -f olimex-jtag-tiny.cfg -f stm32.cfg -f  
stm32_gdb.cfg
```

Server és JTAG konfiguráció

stm32_gdb.cfg

default ports

telnet_port 4444

gdb_port 3333

tcl_port 6666

init

jtag_khz 565

reset init

verify_ircapture disable

olimex-jtag-tiny.cfg

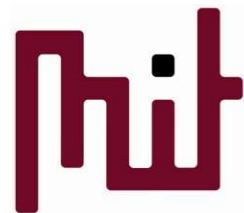
interface ft2232

ft2232_device_desc "Olimex OpenOCD JTAG
TINY"

ft2232_layout olimex-jtag

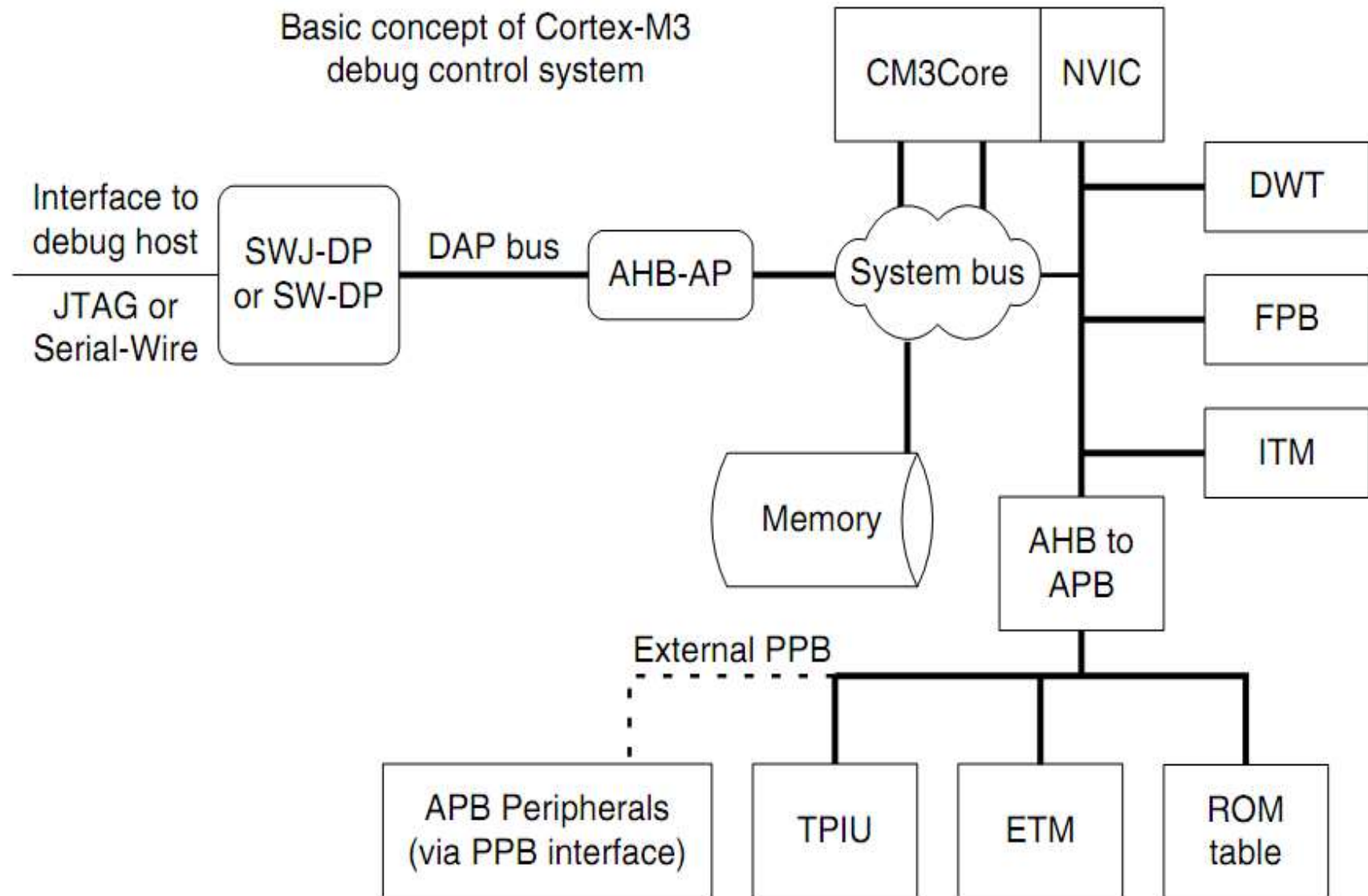
ft2232_vid_pid 0x15ba 0x0004

ARM Cortex M debug képességek



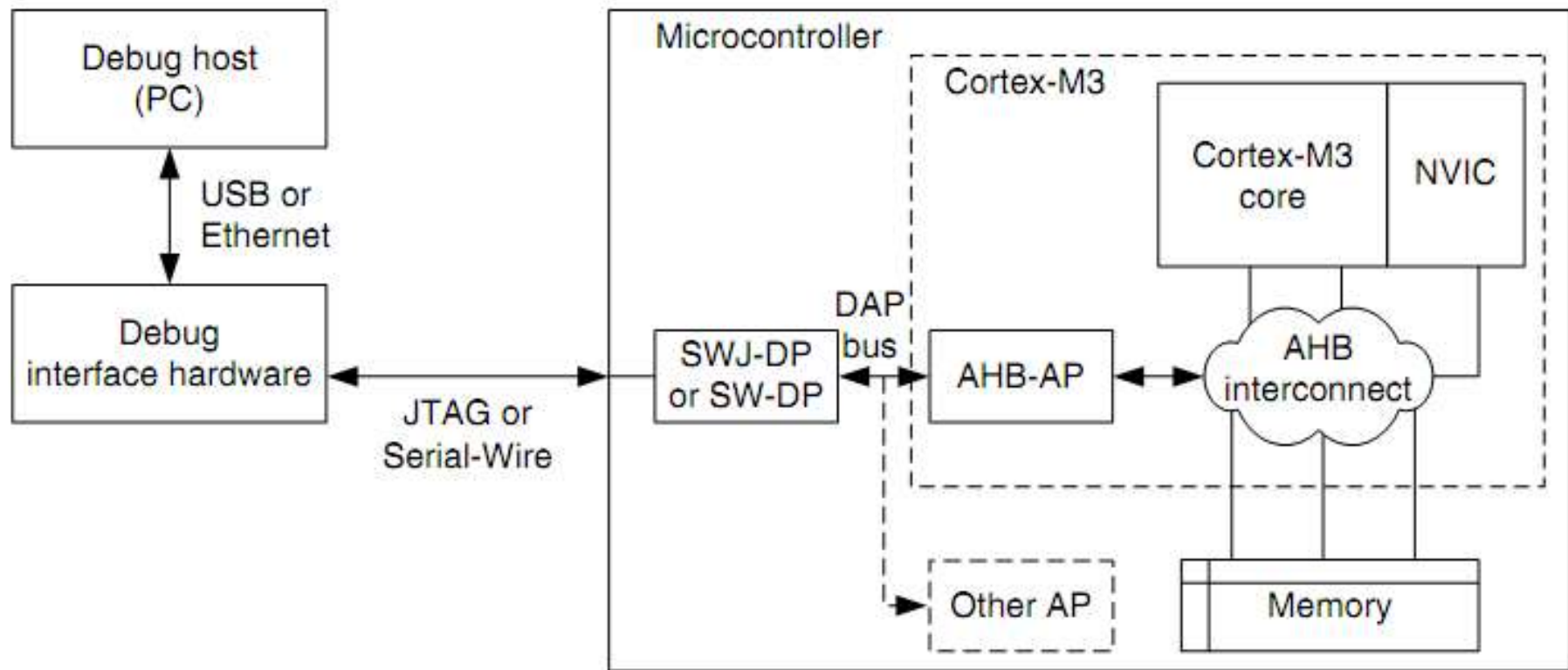
Méréstechnika és
Információs Rendszerek
Tanszék

Cortex M: Coresight debug rendszer



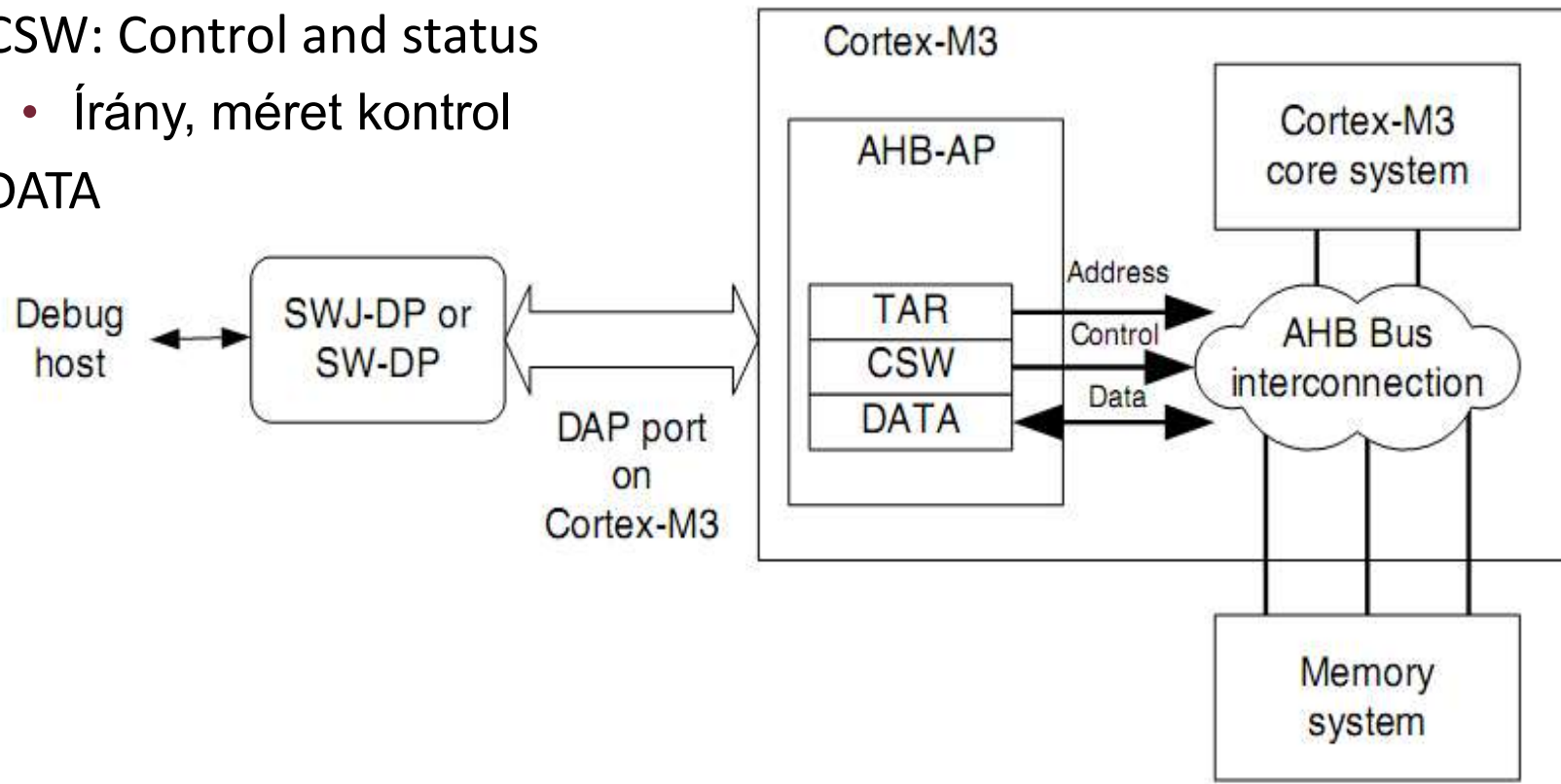
Debug Port csatlakozás

- SWD, vagy JTAG csatlakozás



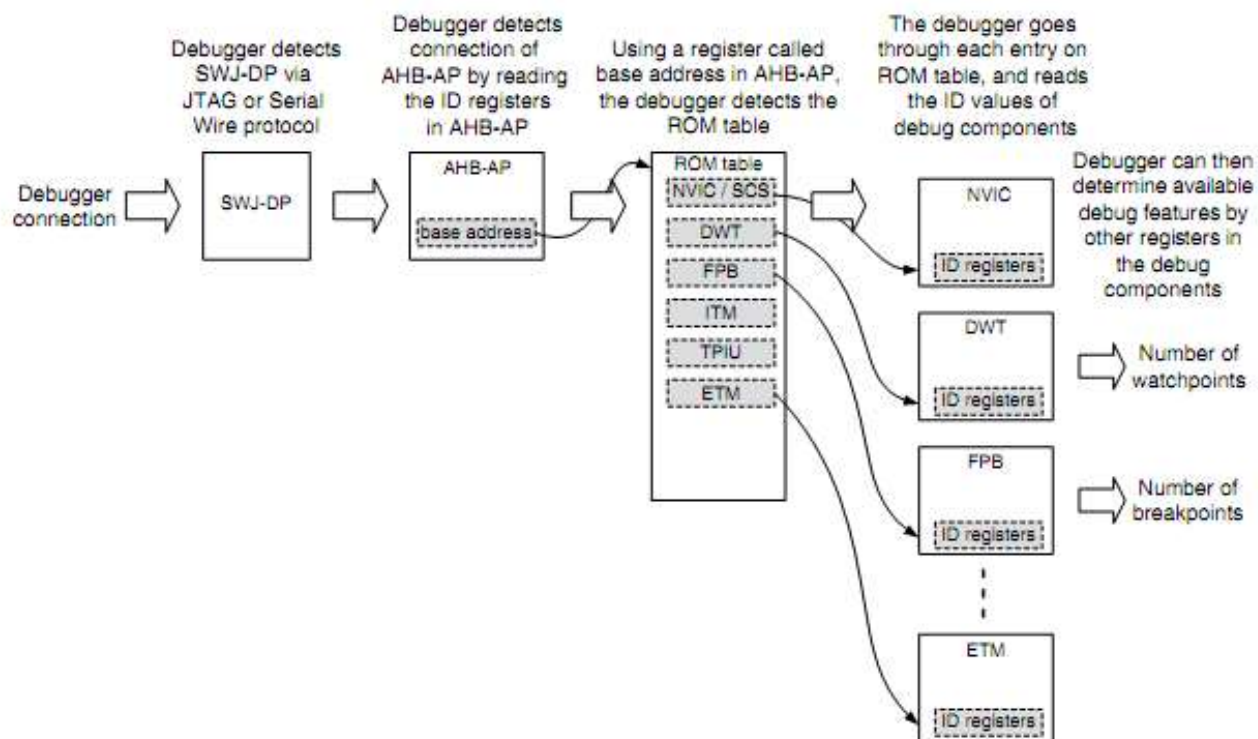
AHB-AP: Advanced High-Performance Bus Access Port

- Bridge a debug portok és a Cortex M3 rendszer között
 - TAR: Transfer Address
 - Átviteli cím kontrol
 - CSW: Control and status
 - Írány, méret kontrol
 - DATA



Debug feltérképezési folyamat

- Van egy ún. ROM table ami tartalmazza a komponensek címeit.
 - Alapból egységes komponensek, de az új sorozatokban bővíthették őket



Debug Módok

- C_DEBUGEN bit a Debug Halting Control and Status register-ben, hogy debug módba kerüljön a processzor
 - Csak a DAP-on keresztül állítható (külső eszköz)
 - A Halt-ba rakás is ez a regiszter, de az már SW-ből is állítható
- 1. Halt mód
 - Utasítás végrehajtás leállítódik
 - A System Tick Timer (SYSTICK) counter leáll
 - Step utasítások
 - Interuptok felfüggesztődnek, és a léptetés alatt vagy végrehajtnak, vagy lemaszkolhatóak

Debug Módok

- 2. Debug monitor mode
 - A processzor végrehajtja a 12. kivételt (debug monitor)
 - A SYSTICK counter tovább fut
 - Az új bejövő interruptok vagy megszakítják, vagy nem a debug monitort annak prioritásának függvényében
 - Debug esemény elveszhet, ha magasabb prioritású IT hajtódik végre
 - Támogatja a single-step működést
 - Memória tartalom megváltoztatható a debug monitor handler-en keresztül

Debug Módok

- 2. Debug monitor mode
 - A processzor végrehajtja a 12. kivételt (debug monitor)
 - A SYSTICK counter tovább fut
 - Az új bejövő interruptok vagy megszakítják, vagy nem a debug monitort annak prioritásának függvényében
 - Debug esemény elveszhet, ha magasabb prioritású IT hajtódik végre
 - Támogatja a single-step működést
 - Memória tartalom megváltoztatható a debug monitor handler-en keresztül
- Azokhoz a rendszerekhez, ahol nem engedhető meg a teljes rendszer leállítása (motor control, disk vezérlés)
 - Debugger tudja a thread szintet és az alacsony prioritású IT-eket figyelni, a magas prioritású IT meg működik zavartalanul.

FPB: Flash Patch and Breakpoint Unit

- Hardware-es breakpointok generálása
 - 8 komparátor
 - 6 program cím
 - 2 literál
- Flash Patch feature
 - Módosítások a nem módosítható Flash kódon
 - 2 literál komparátor
 - Flash területeket remappell az SRAM-ba
 - Számunkra nem lényeges, csak ROM alapú eszközöknél

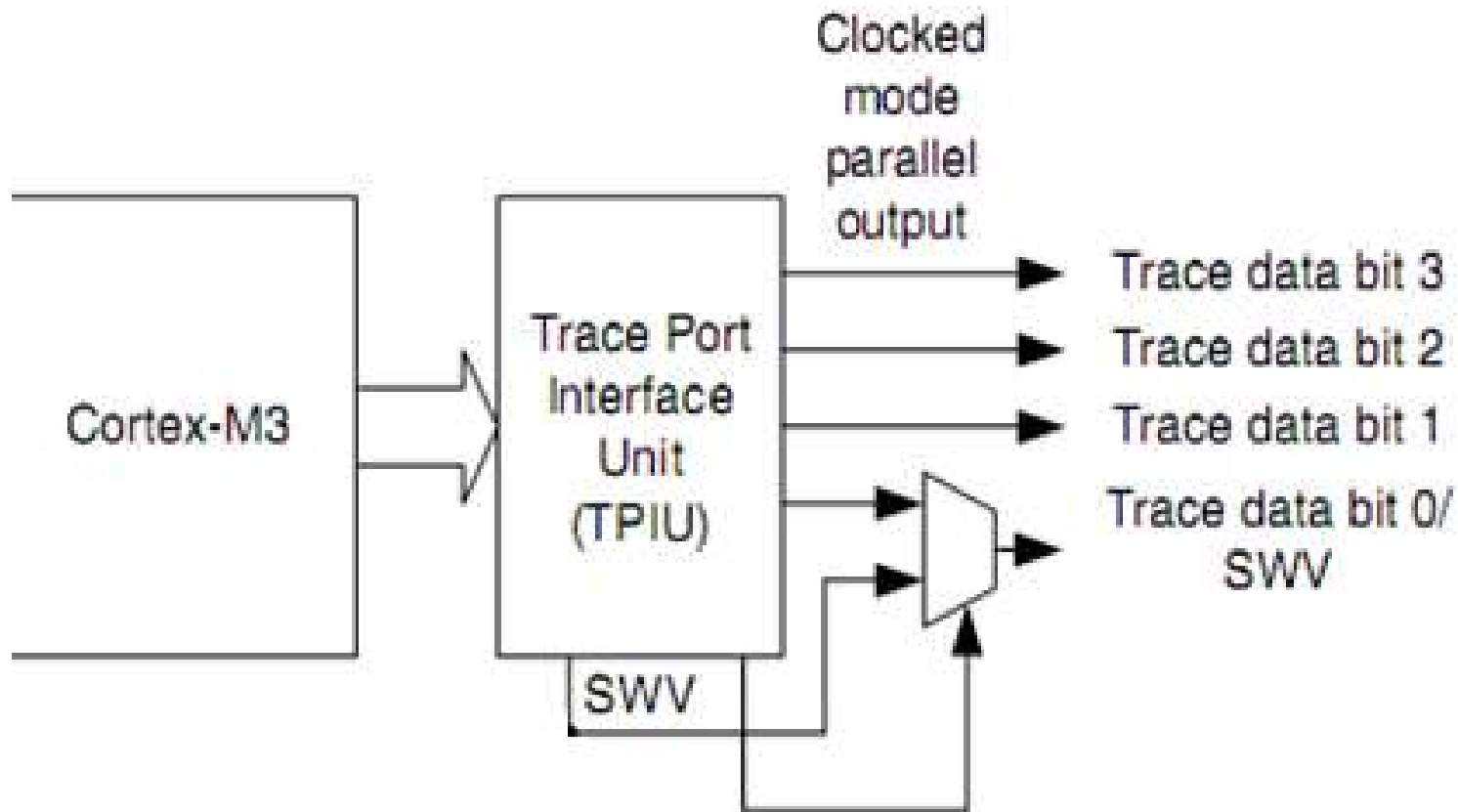
ARM Cortex M trace képességek



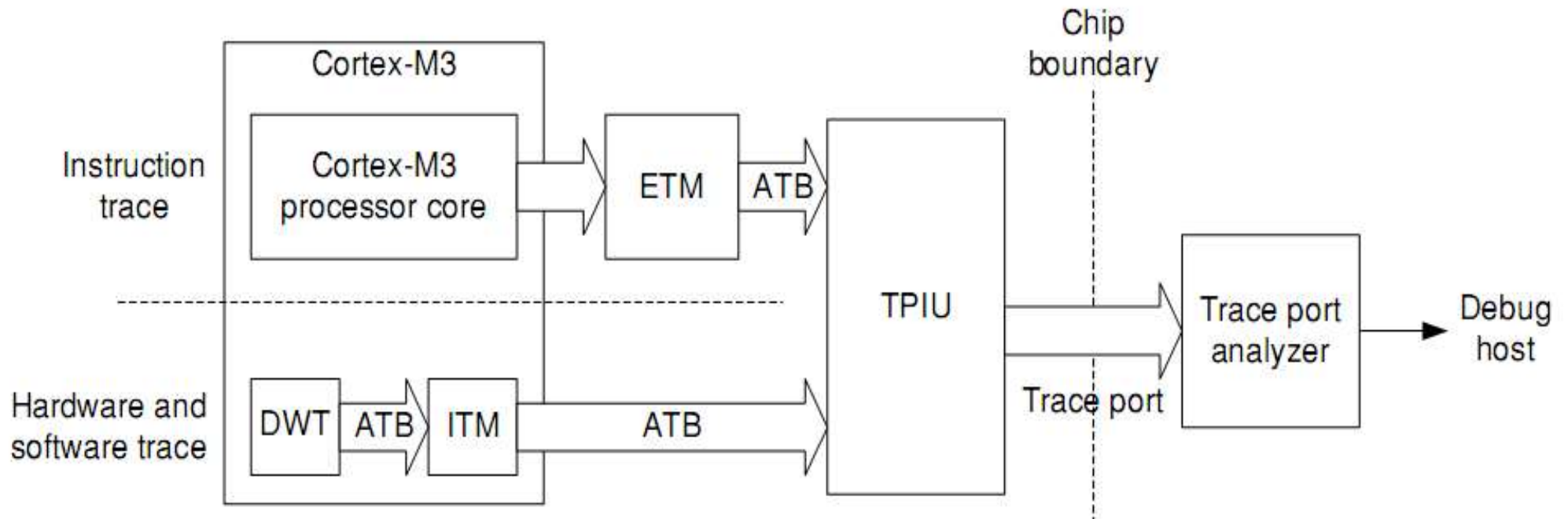
Méréstechnika és
Információs Rendszerek
Tanszék

Trace portok

- 1-4 vezetékes szinkron átvitel a processzor órajelének felével DDR (double data rate) átvitelrel
- 1 vezetékes aszinkron UART jellegű átvitel



Coresight trace rendszer

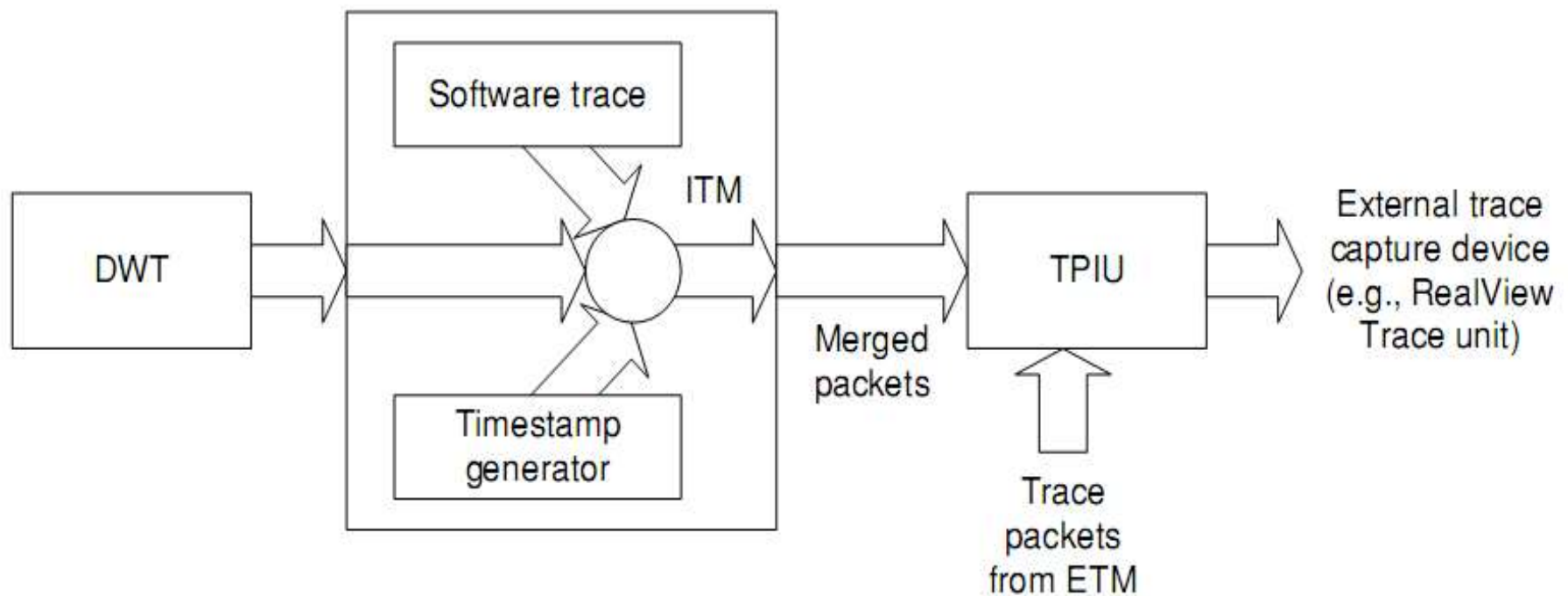


DWT: Data Watchpoint Trace

- 4 komparátor: data address / program counter (az első konfigurálható clock cycles counterre is)
Van hozzá MASK is
 - Hardware watchpoint: processzor debug módba
 - ETM trigger: trace csomag küldés indítás
 - PC mintavételező trigger
 - Data address sample trigger
- Számlálók
 - Órajel számláló
 - Sleep ciklusokat számláló
 - Interrupt overhead számláló
- PC mintavételezés időközönként
- Interrupt trace

ITM: Instrumentation Trace Macrocell

- Direkt konzol üzenetek (printf)
- DWT tud üzeneteket generálni
- Időbélyeg generálás a debugger számára

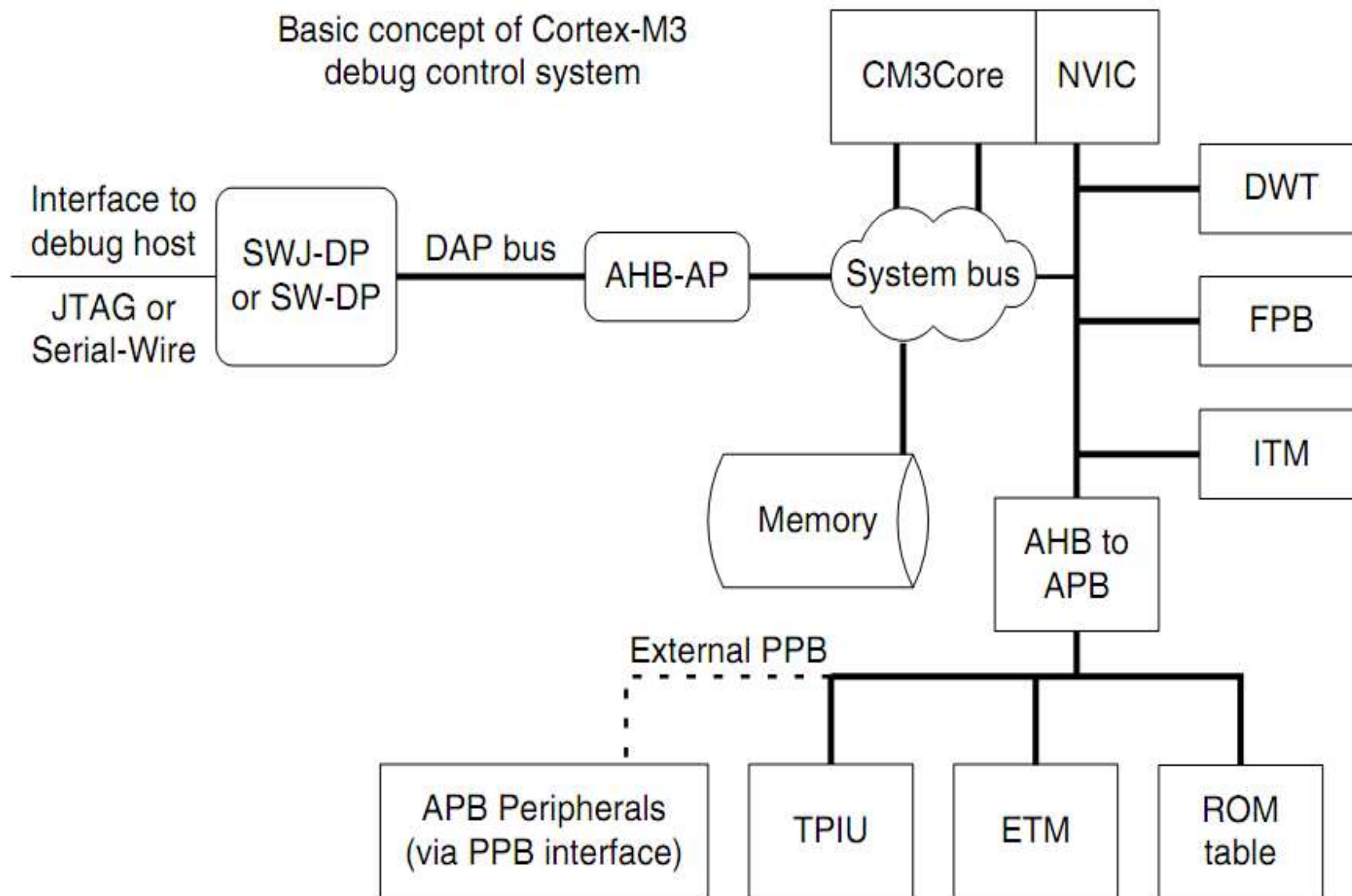


ETM: Embedded Trace Macrocell

- Már az ARM7-eseknél volt
- Utasítás végrehajtás követés
- DWT-et használja komparátorként

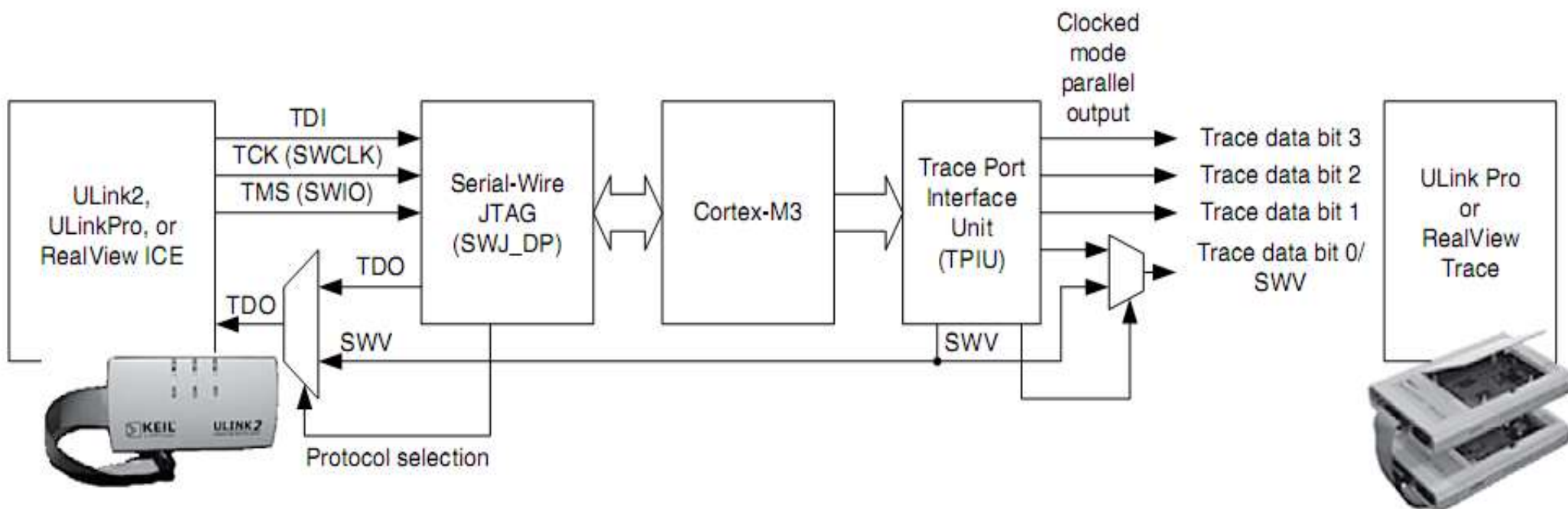
- Kivételek nyomon követése
- Utasítás nyomon követés
 - Gyakorlatilag az összes végrehajtott utasítás nyomon követhető
 - Debuggernek meg kell hogy legyen a bináris kód

Cortex M3: Coresight debug rendszer



TPIU: Trace Port Interface Unit

- 4 bites szinkron mód
 - Elég csúnya formátum
- 1 bites UART szerű aszinkron mód



TPIU: Trace Port Interface Unit

- Trace kimenet

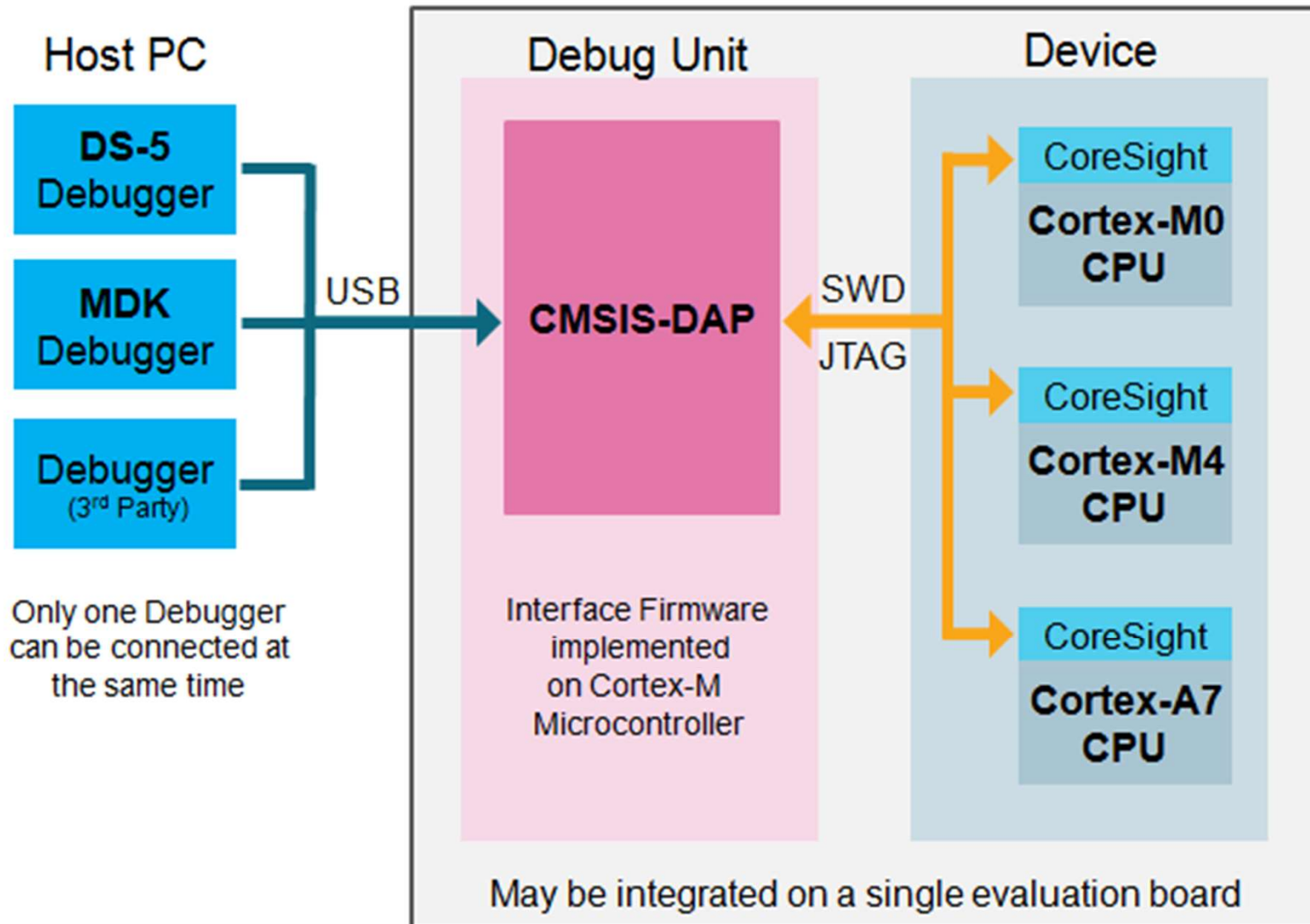
	31	24	23	17	16	15	8	7	1	0	
Bytes 3-0	Data 0xA7				Data 0x53		0	Data 0xAA		ID 0x03	1
Bytes 7-4	Data 0x52				Data 0x2A		0	Data 0xA8		ID 0x15	1
Bytes 11-8	Data 0xCA				ID 0x03		1	Data 0x54		Data 0x29	0
Bytes 15-12	0	0	0	1	1	1	0	0	Data 0x64		0
					Data 0x64		0	Data 0xC7		Data 0x63	0

Az új generáció CMSIS-DAP

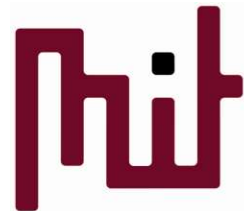


Méréstechnika és
Információs Rendszerek
Tanszék

CMSIS-DAP



Az új generációs debug, trace kéességek alkalmazása



Méréstechnika és
Információs Rendszerek
Tanszék

Hardware-In-The-Loop tesztek

A beágyazott vezérlő egy a valóságot szimuláló környezetben hajtja végre a funkcionalitását

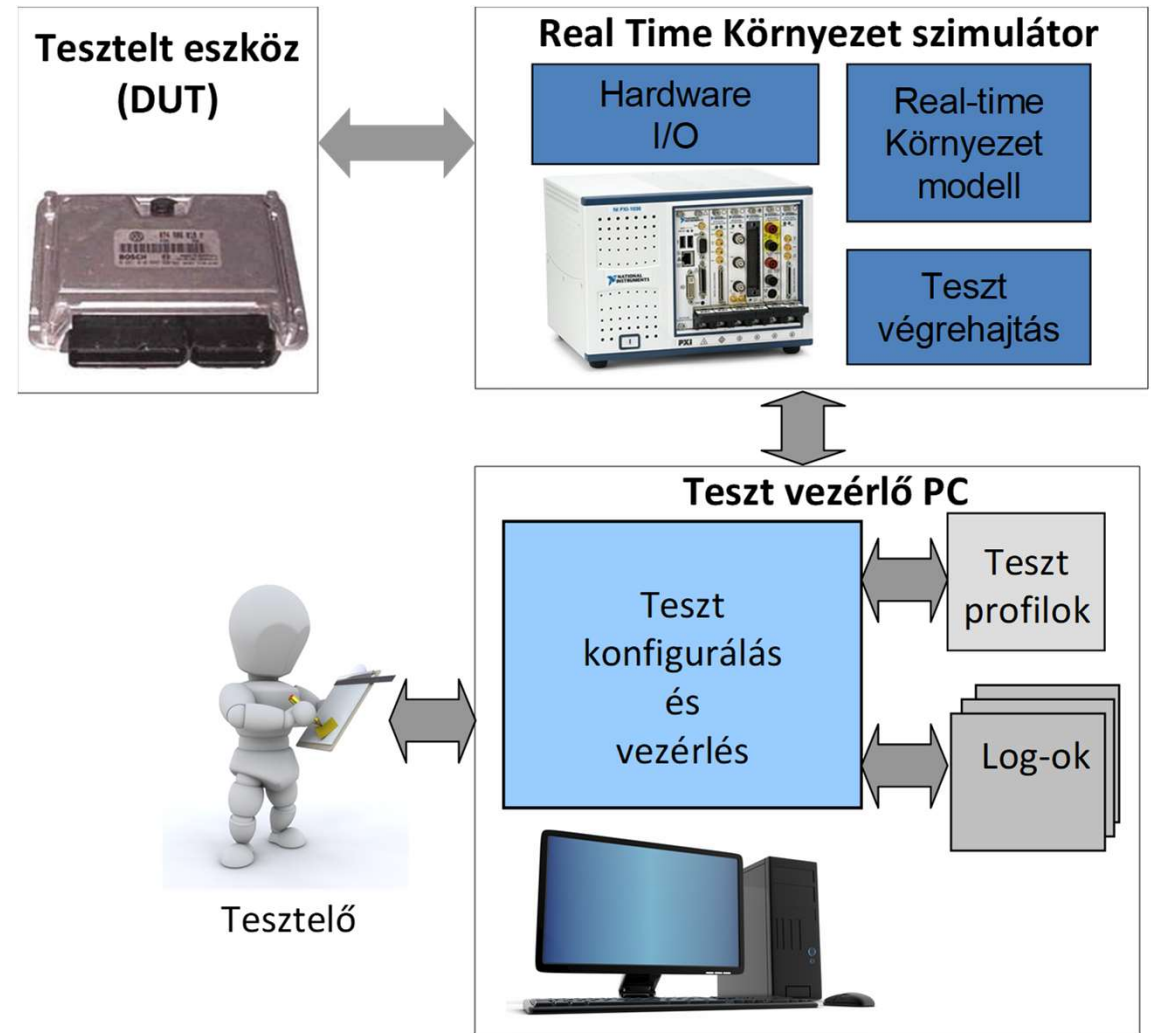
- Könnyű felműszerezés
- Megismételhető tesztek
- Nincs veszélyben a tesztelő a tesztelt eszköz és a környezet sem



Forrás: ni.com

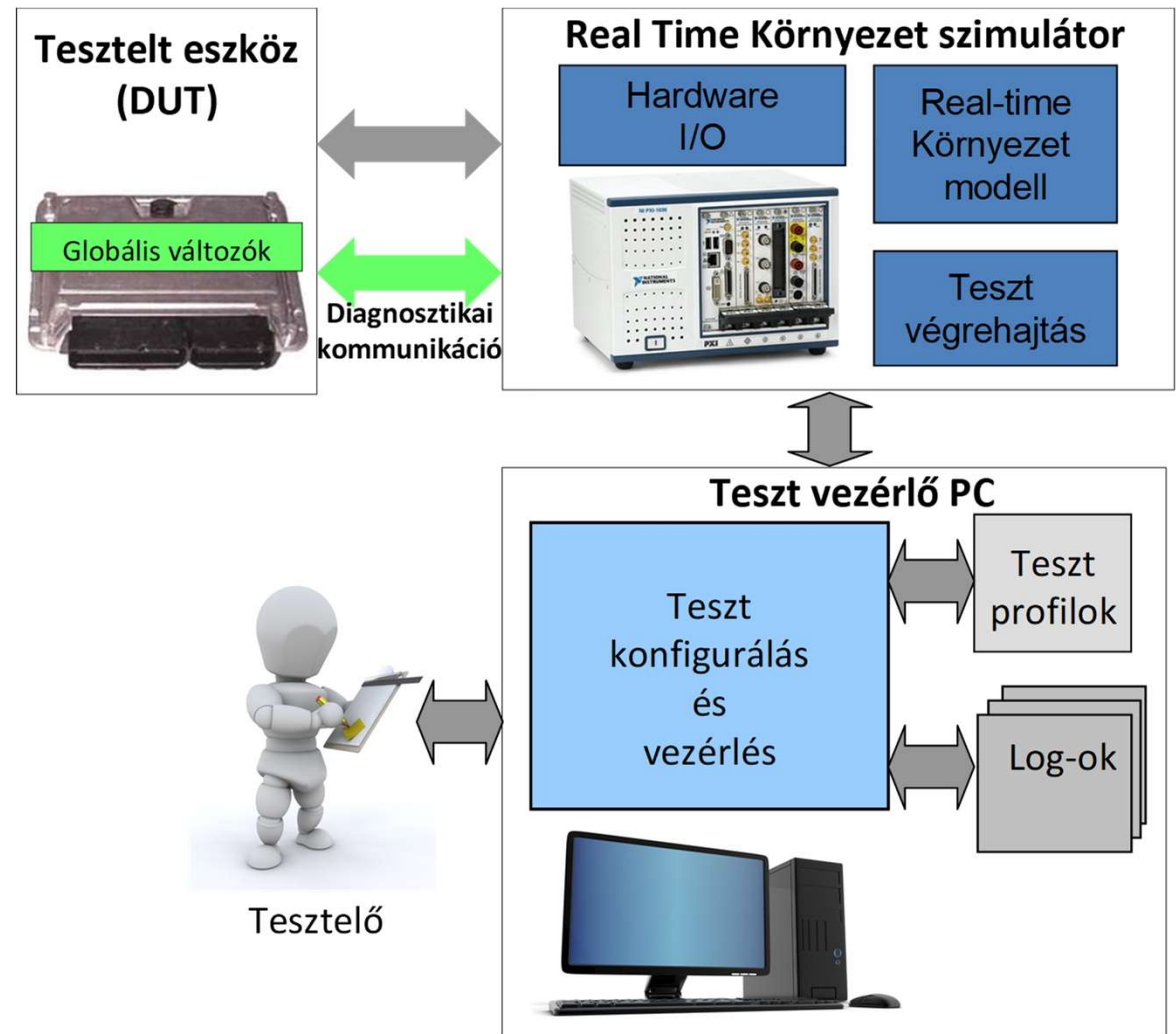
Egy tipikus HIL teszt környezet

- **Teszt adatok**
 - Analóg I/O
 - Digitális I/O
 - Kommunikációs



Egy tipikus HIL teszt környezet

- **Teszt adatok**
 - Analóg I/O
 - Digitális I/O
 - Kommunikációs
- **Másodlagos információk**
 - Rendszer belső működéséről
 - Globális változók



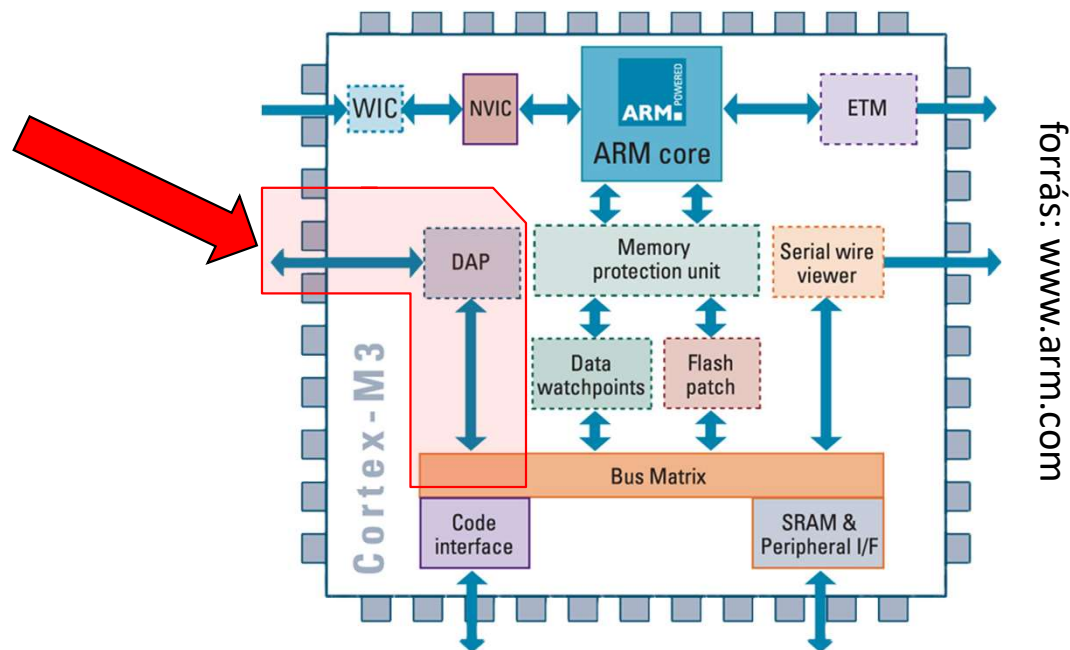
Másodlagos adatok kinyerése

- Diagnosztikai kommunikáció
 - Előnyök
 - Kész protokollok: CCP, XCP, UDS, KWP2000
 - Kész toolok: NI, Vector CANape, ETAS INCA
 - Hátrányok
 - Megosztott kommunikációs közeg
 - Limitált adatátviteli sebesség
 - Memória foglalás
 - Processzor erőforrás foglalás
- Hardware támogatott megoldások
- Szükségesek leíró file-ok, amik a globális változók tulajdonságait tárolják pl: A2L file
(ASAM MCD-2 MC)

Modern mikrovezérlők debug képességei

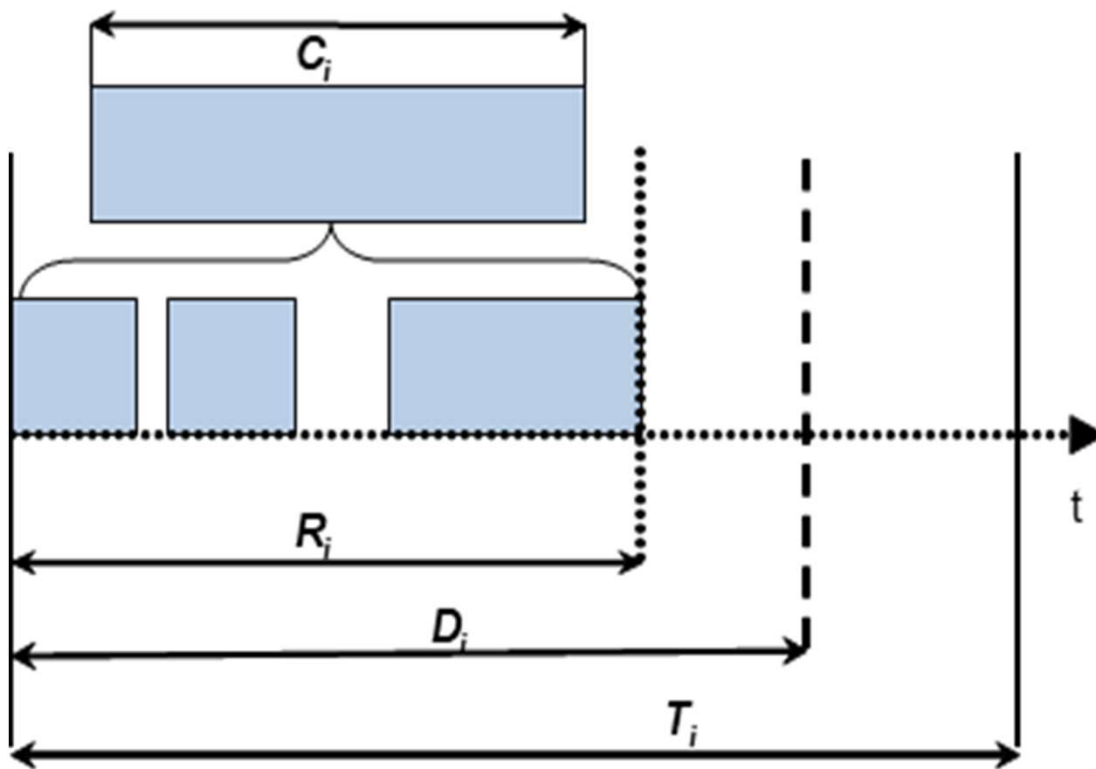
- ARM magú vezérlők CoreSight szabvány alapú támogatás, más vezérlők IEEE-ISTO 5001-2003 (Nexus)
- Non-intrusive memória hozzáférés (CoreSight (AHB-AP), Nexus Class 3+)
 - Memória tartományok kiolvasása és írása futás közben

Debug hozzáférés a teljes belső buszmátrixhoz



Foundations of Timing and Synchronization failure detection

- Az ütemezhetőség számításának alapja a WCETs (Worst Case Execution Time) számítás, vagy mérés --> Worst Case Response time számítható a WCET-ből

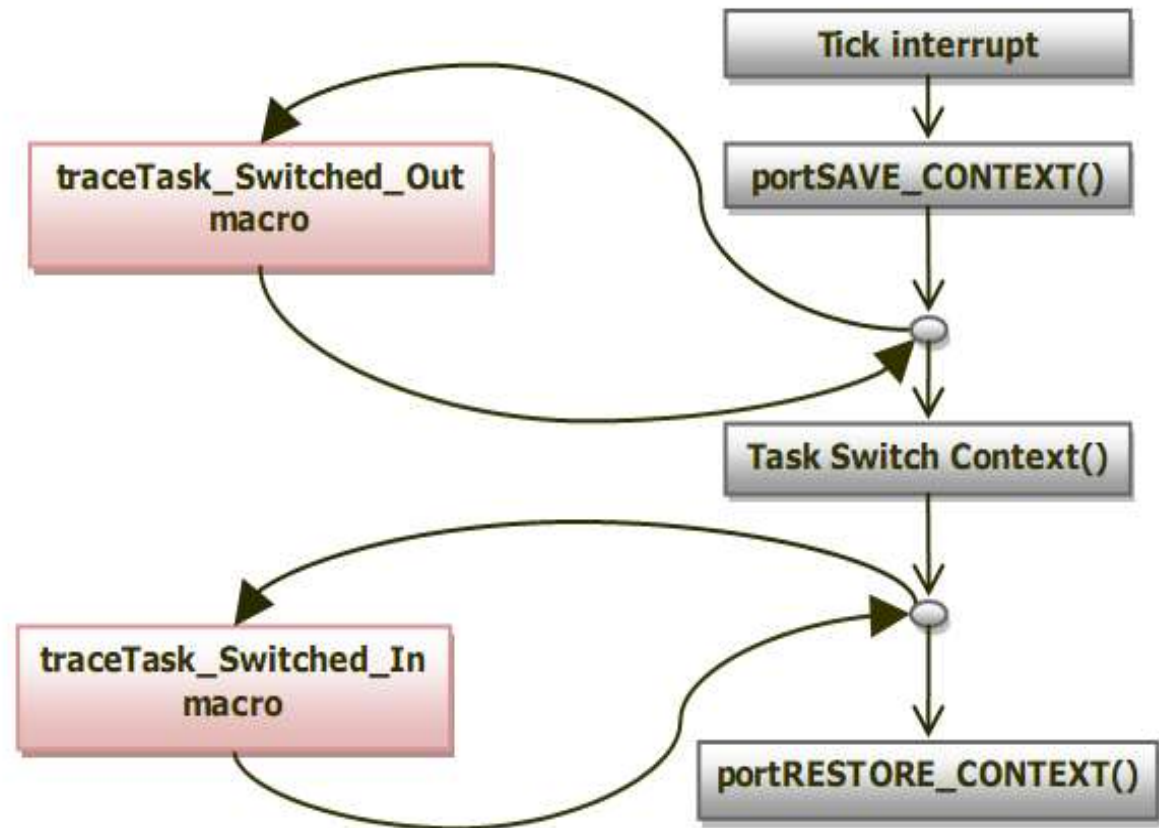


T_i is the period of task i
 D_i is the deadline of task i
 C_i is the WCET time of task
 R_i is the worst-case response
time to task i

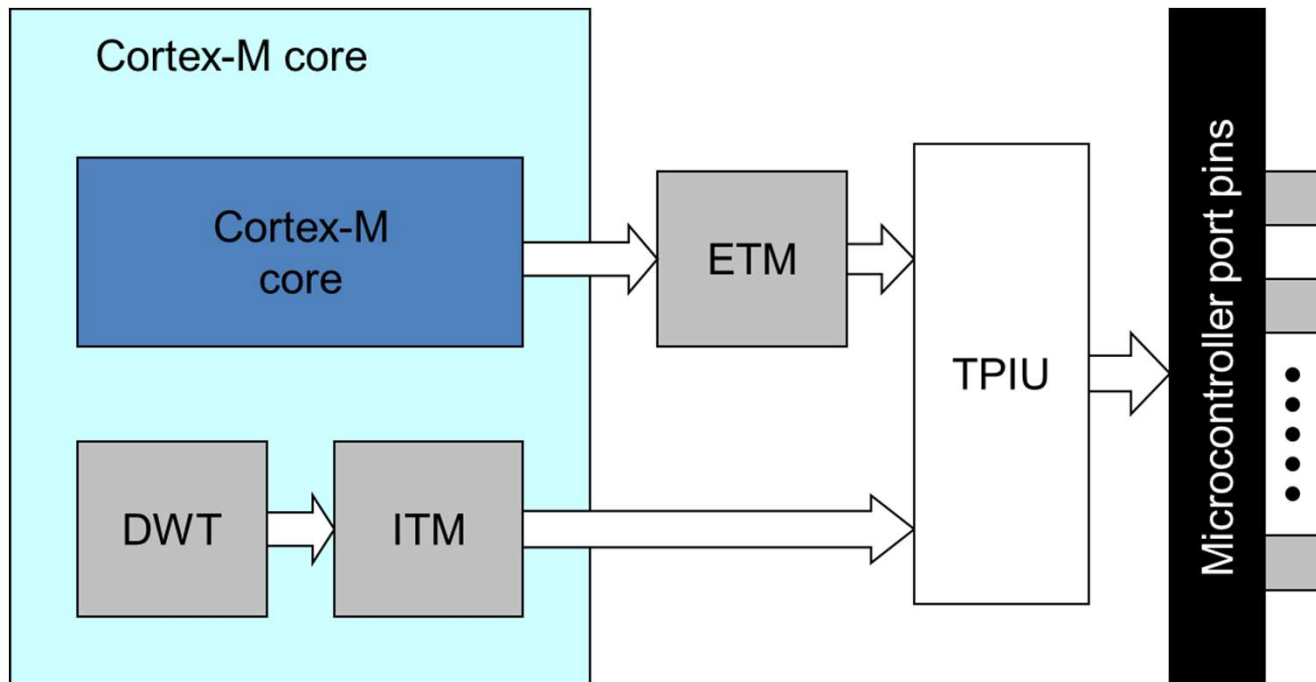
A task execution time számításának elméleti módjai

■ RTOS műszerezés alapú megoldás

- Mély kernel szintű műszerezés szükséges
- Néhány kernel támogatja ezt
- Vannak erre épülő profi megoldások
- Kód méret és végrehajtási idő overhead-et okoz
- Intenzív kernel – teszt rendszer kommunikációt
- követel



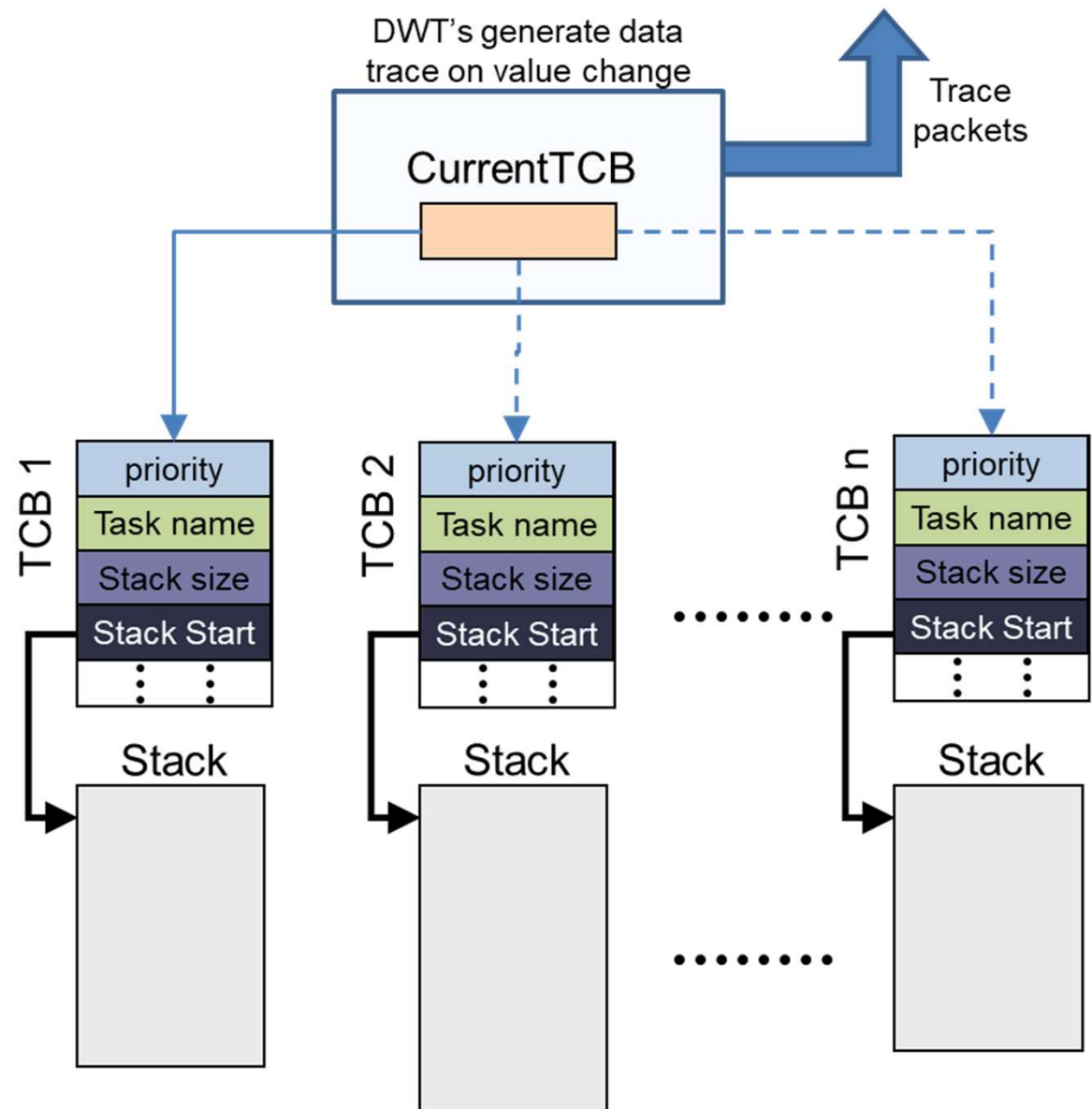
Trace alapú mérések



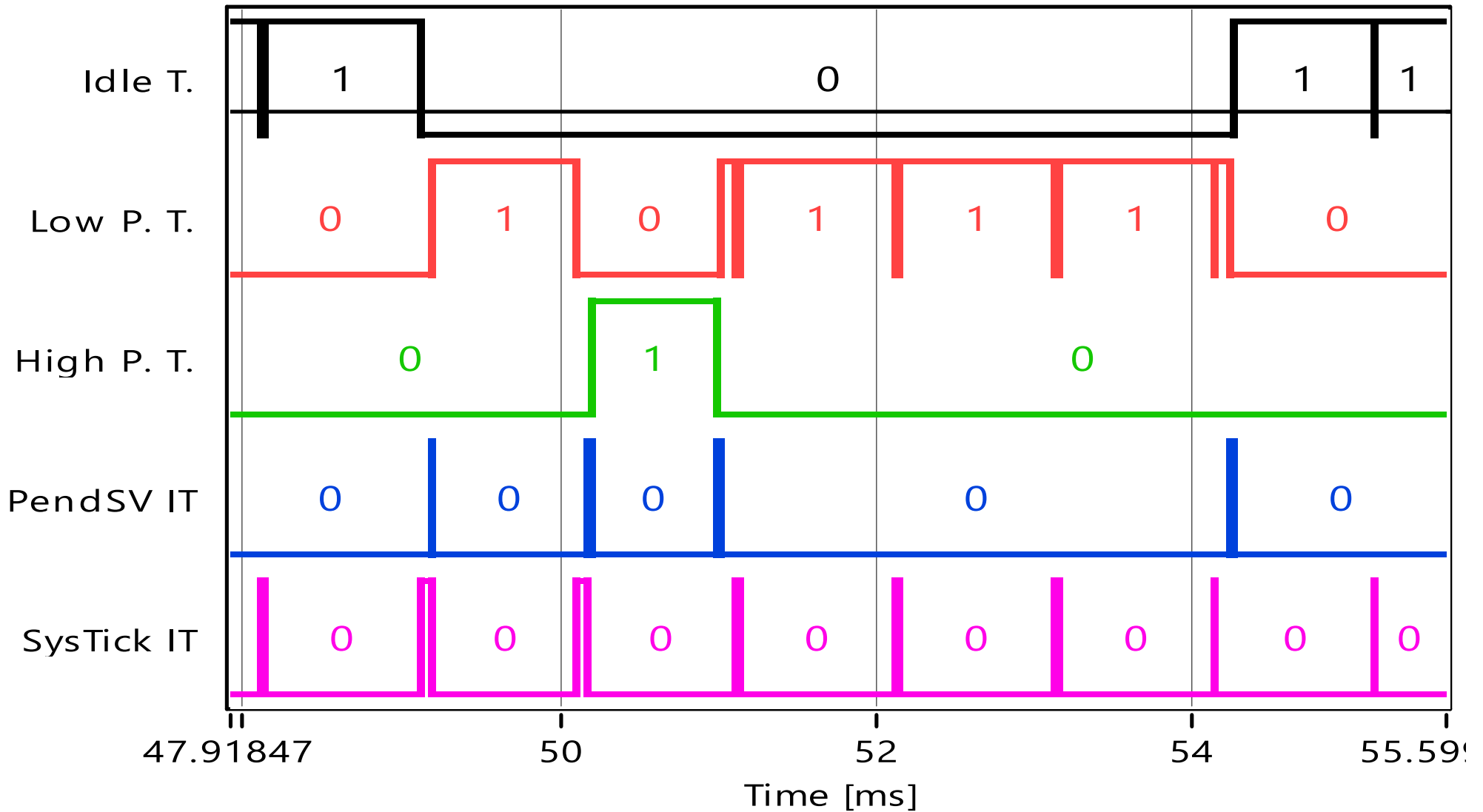
- *ITM (Instrumentation Trace Macrocell) és DWT képességek elegendek hozzá*

A CurrentTCB változását kell figyelni

- Watch point a CurrentTCB-re
- Interrupt overhead is mérhető
- 1 vezetékes SWV funkcionalitás elég lehet

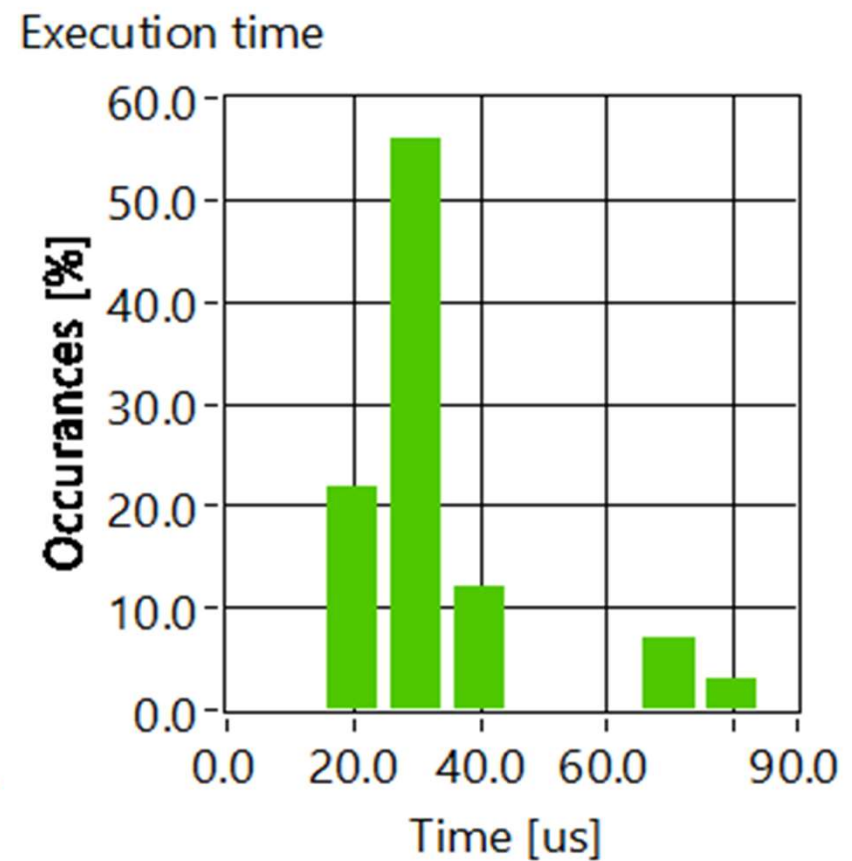
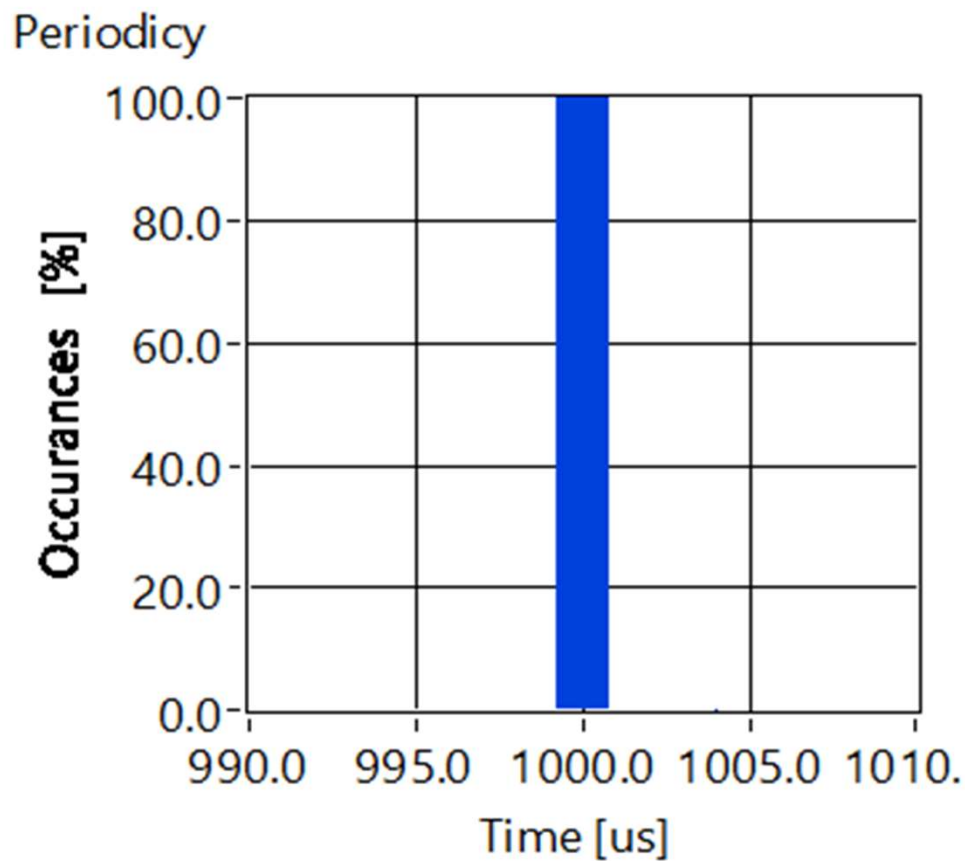


Minta a taszk ütemezhetőség mérésről



IT periodicitás és végrehajtási idő mérés

- Periodikus IT, min a heart beat timer IT



Taszk végrehajtási idő mérés

- Minta egy alacsony prioritású taszkra

Execution time

