

多核调试

1. 多核调试的好处

在 IAR EWARM 的 7.10 版本，我们引入了多核设备同步调试的功能。虽然多核调试比较复杂，且我们会出一个系列的文章来介绍，但其实使用 IAR 的新版本来调试多核相对来说比较的简单，在这篇文章中我将告诉你更多关于通用多核调试，特别是用 IAR EWARM 如何处理多核调试的内容。

2. 为什么需要多核调试

现在，复杂应用的需求正在增加，多核设备变得越来越流行是一个明显的趋势。通过使用多核设备，采取不同强度的不同架构可以平衡负载，使用一个多核设备代替多个设备可以降低产品成本。

在一个多核环境中，调试一个单核已经不是问题了。但我们现在介绍的是多个内核同步调试，有了同时调试 2 个或多个内核的能力，你就会更容易找到错误和故障。在某些情况下，当其它核心可能会影响你正在使用的核心时，在你现在不关心的核上至少可以处理像启动、停止和监控这些基础任务也是一个优势。

多核调试的主要挑战是使用一个接口访问双核。在 IAR Embedded Workbench 中，您可以同时调试两个或两个以上相同的内核[对称多核处理(SMP)]，或使用工具链的一个 Master 实例和 Slave 实例调试不同结构的两个核[非对称多核处理(AMP)]。这两个内核可以从每个实例单独或同时启动和停止。这都是在单独的开发环境中完成的。

3. 两种不同的使用情况

我们可以看到两种不同的使用情况，一个是在同一类型的多个内核上进行调试(SMP)，另一个是不同类型的多核上进行调试(AMP)。对称多核调试是使用调试器的一个实例进行的，而在非对称的情况下，每个核要求集成开发环境调试器的一个实例，一个扮演 Master，另一个扮演 Slave。

3.1 异步多核调试

在非对称多核处理(AMP)，例如：你可以使用双 ARM Cortex-A9 芯片或一个 Cortex-M 核和一个 Cortex-A 核的芯片。在下面的示例中，Cortex-A 核的项目设置作为一个 Master，该项目还将下载 Slave 核的代码到共用存储器中。Slave 项目里唯一设置改变的是不下载代码，只连接到设备。

当启动 Master 调试器界面时，它将启动从调试器界面，如下图一样：

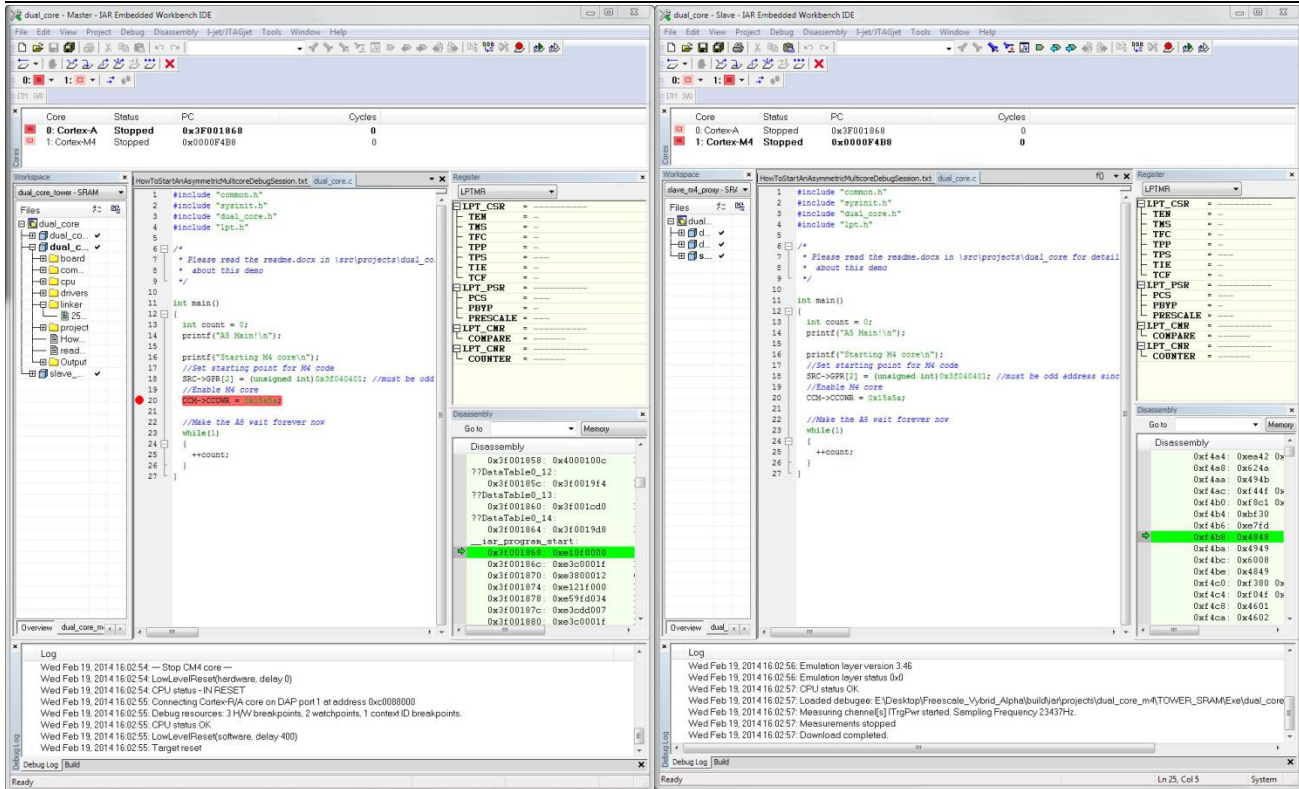


图 3-1

与传统单核界面不同:

- 每一个调试窗口都是 Master 或 Slave
- 一个新的 Core 调试窗口
- “调试器”工具栏中有一些新的选项

如果我们仔细看看在集成开发环境中新的有趣的选项，第一个是核(Core)的窗口。

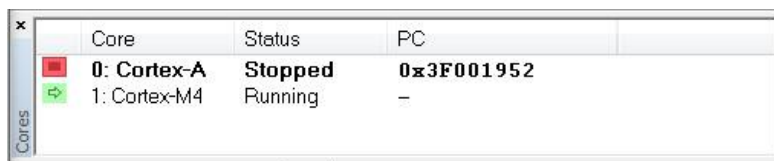


图 3-2

在这个窗口中，我们可以看到集成开发环境的实例，连接到 Cortex-A 核（粗体），它停了下来，并且还显示了核的 PC。我们还可以得到 Cortex-M 核的一些信息，即它是运行状态的。

另一个是新项目对调试工具条的补充:



图 3-3

首先，由 2 个图标描述的核心状态，然后新的控制启动和停止所有核的按钮。

通过在工具栏中的核之一上停留，可以得到进一步的信息，匹配核窗口中的信息。在这个工具栏中，也有为每个核提供的一个下拉菜单，并且使用这些下拉菜单中的选项，您可以停止或启动选定的核。

3.2 对称多核调试

调试一个带有多个内核的对称设备，类似于上面的例子。但是代替两个实例，只有一个集成开发环境实例显示信息。在这里，在核窗口中，选择你使用的核，如观察窗口和内存窗口等正常的调试窗口将显示你正在使用的相关信息。

4. 做正确的设置

有一些特殊的设置要进行，但是不多。例如，在非对称的情况下，对于 Master 调试界面，您必须在其中声明其为多核 Master，并为其使能一个 Slave“使用”的项目。对于 Slave，我们已经提到，主程序调试界面中可以将代码下载到设备，而 Slave 程序调试界面不能做到这一点。还有选择复位策略不影响 Master 程序的调试，所以通常在 Slave 调试界面中使用某种软件复位。

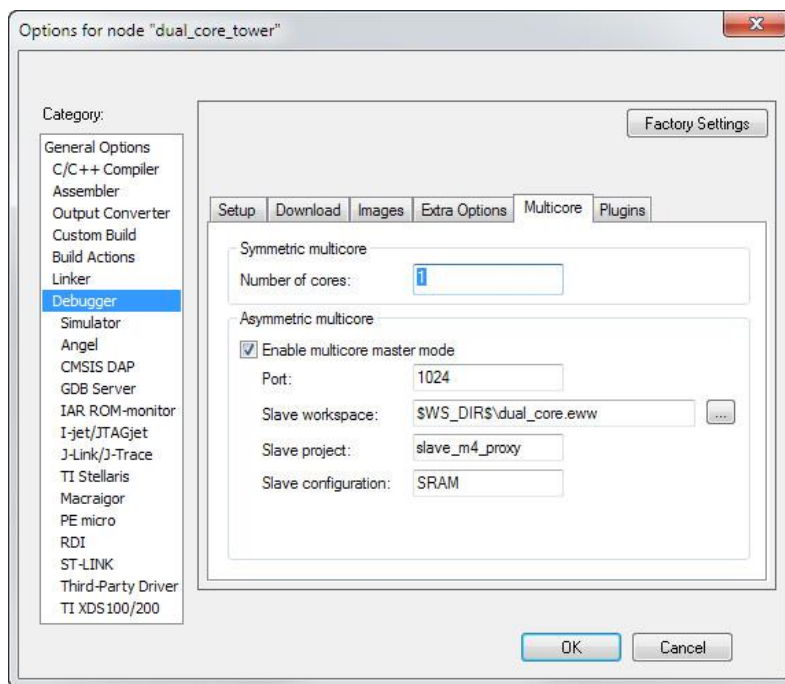


图 4-1

5. 小结

一旦你已经进行了这些设置，老实说，大部分的问题还是会存在，多核调试不会比单核调试更容易。如果你之前只使用过调试单核的工具，你可能永远不会了解其他人谈论的多核调试有多么困难。