OpenOCD for AArch64

What is OpenOCD?
From http://elinux.org/OpenOCD

"OpenOCD (Open On-Chip Debugger) is open-source software that interfaces with a hardware debugger's JTAG port. OpenOCD provides debugging and in-system programming for embedded target devices. OpenOCD provides the ability to flash NAND and NOR FLASH memory devices that are attached to the processor on the target system. Flash programming is supported for external CFI compatible flashes (Intel and AMD/Spansion command set) and several internal flashes (LPC2000, AT91SAM7, STR7x, STR9x, LM3 and STM32x)."

Stalled AArch64 Port
Presently OpenOCD only supports 32-bit hardware targets, and this includes a number of 32-bit ARM targets. There is an effort to support 64-bit ARM (AArch64) targets in OpenOCD:
http://openocd.zylin.com/#/c/2523/
Unfortunately this patch has not yet been accepted into the upstream OpenOCD project and, in-fact, has stalled and not made any forward project in over 1 year.

Necessary Steps to Restart Stalled AArch64 Port
This patch is a good starting point for AArch64 OpenOCD, however, there are still some problems to be address in the patch set:

1. Need to address some pervasive 'style' issues to make it comply with existing OpenOCD style and best-practices.
2. The patch needs to be split into several parts so that upstreaming can be staged and bike-shedding on some topics doesn't delay upstreaming of non-controversial code.
3. OpenOCD support for 64-bit targets needs quite a bit of work. OpenOCD assumes the target address is 32-bits. There is a pending patch for 64-bit OpenOCD support:
http://openocd.zylin.com/#/c/1200/
4. GDB support for the current AArch64 OpenOCD prototype is questionable. The original patch author has said that gdb does not work with his patch-set. This might point to a problem in either GDB or in OpenOCD.
5. There needs to be code-sharing with the 32-bit Cortex-A backend so that duplication is eliminated (see cortex_a.c).
6. Reviewers have suggested reusing as much of the arm dpm operations code for AArch64 as possible, and abstracting the AArch64 code out into a separate AArch64 variant file when necessary. This remains to be done.

Further Technical Considerations
While all of these topics need to be addressed there are some further considerations for a complete AArch64 port:

1. Proper 64-bit target support in OpenOCD must be committed prior to upstreaming of the AArch64 patch-set. This means that any effort /sizing must include this effort as well.
2. Due to code-reuse it is necessary to setup both an ARM and AArch64 regression testing environment in order to prove that any changes for AArch64 don't regress the ARM target.
3. OpenOCD to GDB and LLDB integration must be tested and enabled (respectively). The GDB test-suite must be run against a target connected with OpenOCD.
4. Must check the license/copyright of the existing patch-sets to verify that it is appropriate to adopt the existing patch-sets.
5. Address OpenOCD's ability to support AArch64 in different ARMv8 exception levels. This will require investigation of the complexity of debugging programs in different exception levels.
6. Investigate the interface of OpenOCD for AArch64 with the Coresight Debug/Trace interface (instead of JTAG).