本实验用到的指令集如下：



在修正代码后

首先，我们要修改func配置文件

编译func程序需要使用LoongArch32R的GCC交叉编译工具

这里我是在Windows下运行的Vivado

而该工具需要在Linux操作系统下使用

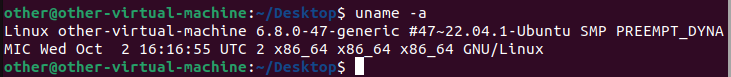
那么

我们准备一个Linux虚拟机，这里用的是Ubuntu 22.04

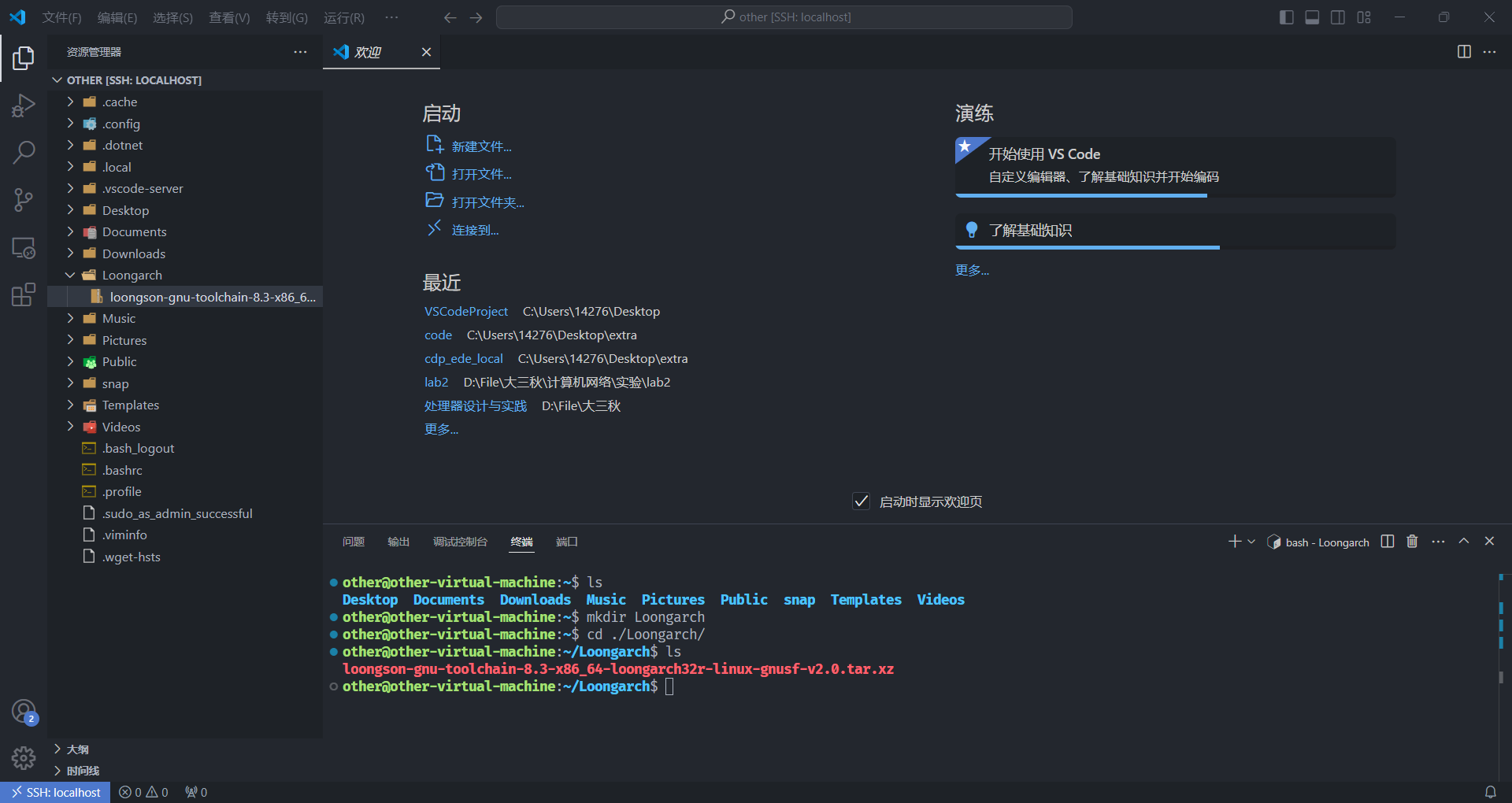
电脑软件的截图

描述已自动生成

如果是x86架构，需要是64位系统



从<https://gitee.com/loongson-edu/la32r-toolchains/releases> 中下载对应的LoongArch32R的GCC交叉编译工具，存放至虚拟机中



这里我下载的是loongson-gnu-toolchain-8.3-x86\_64-loongarch32r-linux-gnusf-v2.0.tar.xz

在该压缩包所在目录，进行解压：

sudo tar xvJf loongson-gnu-toolchain-8.3-x86\_64-loongarch32r-linux-gnusf-v2.0.tar.xz -C /opt/

这里我们确认/opt/loongson-gnu-toolchain-8.3-x86\_64-loongarch32r-linux-gnusf-v2.0/bin/目录确实存在

图形用户界面, 文本, 应用程序, 聊天或短信

描述已自动生成

随后执行：

echo "export PATH=/opt/loongson-gnu-toolchain-8.3-x86\_64-loongarch32r-linux-gnusf-v2.0/bin/:$PATH" >> ~/.bashrc

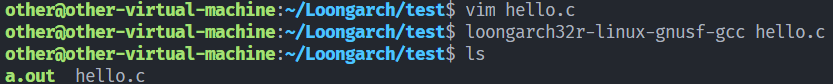
执行source ~/.bashrc，或重开一个终端，使bashrc生效

发现能够找到如下命令，说明工具链配置完成

文本

描述已自动生成

可编写一个hello.c文件进行测试



接下来

将…\cdp\_ede\_local\mycpu\_env\func文件夹共享至我们的虚拟机

共享后应处于/mnt/hgfs目录中

文本

描述已自动生成

进入func目录，使用make EXP=6编译

文本

描述已自动生成

编译结果含义如下

图形用户界面, 文本, 应用程序

描述已自动生成

打开gettrace工程，即…/cdp\_ede\_local/mycpu\_env/gettrace/gettrace.xpr

运行gettrace工程的仿真（进入仿真界面后，直接点击run all等待仿真运行完成），生成新的参考trace文件golden\_trace.txt

文本

描述已自动生成

打开Vivado 2019

点击界面下方的Tcl Console

输入命令：

cd …(用于存放clone仓库的路径)/cdp\_ede\_local/mycpu\_env/soc\_verify/soc\_dram/run\_vivado

进入create\_project.tcl所在目录

输入命令：

source ./create\_project.tcl

根据create\_project.tcl创建Vivado项目

更新代码文件至…(用于存放clone仓库的路径)/cdp\_ede\_local/mycpu\_env/myCPU/

双击PROJECT MANAGER的Source框中所使用的inst\_ram

目录树：Design Sources > soc\_lite\_top > inst\_ram

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

重新定制inst\_ram的COE File，选择对应func的coe文件（mycpu\_env/func/obj/inst\_ram.coe）

图形用户界面, 文本, 应用程序

描述已自动生成

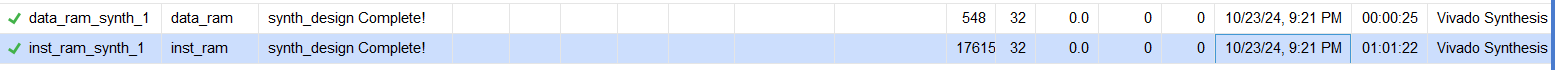
同理，更新data\_ram

图形用户界面, 文本, 应用程序

描述已自动生成

（需要进行Generate操作）

inst\_ram的综合需要比较长的时间，耐心等待



本人机器i5-12500h，内存32GB，使用16jobs，用时约1h

之后可进行仿真，正确日志如下：

图形用户界面, 文本, 应用程序

描述已自动生成

最后进行上机，结果如下：

电子器材

中度可信度描述已自动生成