[\*\* am335x中uboot的编译流程](http://blog.chinaunix.net/uid-28458801-id-3492939.html) 2013-02-20 11:34:47

分类： 嵌入式

**PC操作系统：ubuntu11.10**

**开发板：am335x\_evm**

**uboot ：u-boot-2011.09-psp04.06.00.07**

生成所需文件：MLO(SPL),uboot.img，命令如下：

**#make O=am335x am335x\_evm**

**注意：默认uboot的根目录为 ./**

**makefile的执行流程如下：**

**1，生成板子依赖文件：**

**1,首先根据命令判断是否需要生成目录来保存将要生成的所需的各个文件：**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31 | #########################################################################  #@设定编译输出目录@:  #   函数$( origin, variable) 输出的结果是一个字符串，输出结果由变量variable定义的方式决定，  #   若variable在命令行中定义过，则origin函数返回值为"command line"。  #   假若在命令行中执行了“export BUILD\_DIR=/tmp/build”的命令，则“$(origin O)”值为“command line”，  #   而BUILD\_DIR被设置为“/tmp/build”。  #  #   假若在命令行中执行了“make O=am335x am335x\_evm”的命令，则“$(origin O)”值为“command line”，  #   而BUILD\_DIR被设置为“am335x”。  #########################################################################  ifdef O  ifeq ("$(origin O)", "command line")  BUILD\_DIR := $(O)  endif  endif    #判断 BUILD\_DIR 变量是否为空，当前 BUILD\_DIR 为 am335x，条件为真，则 saved-output 为 am335x  ifneq ($(BUILD\_DIR),)  saved-output := $(BUILD\_DIR)    #若${BUILD\_DIR}表示的目录没有定义，则创建该目录。  # Attempt to create a output directory.  $(shell [ -d ${BUILD\_DIR} ] || mkdir -p ${BUILD\_DIR})    #若$(BUILD\_DIR)为创建失败或不存在，则将其赋值为当前目录路径（源代码目录）。  #并检查$(BUILD\_DIR)目录是否存在。  #Pwd命令用以获取当前路径  # Verify if it was successful.  BUILD\_DIR := $(shell cd $(BUILD\_DIR) && /bin/pwd)  $(if $(BUILD\_DIR),,$(error output directory "$(saved-output)" does not exist))  endif # ifneq ($(BUILD\_DIR),) |

此时会在uboot根目录下生成一个 am335x的目录：**./am335x/**

**2，生成板子依赖文件：**

|  |  |
| --- | --- |
| 1  2 | $(obj).boards.depend:   boards.cfg      awk '(NF && $$1 !~ /^#/) { print $$1 ": " $$1 "\_config; $$(MAKE)" }' $< > $@ |

**读取uboot根目录下的 boards.cfg 文件生成 .boards.depend 隐藏文件，该文件位于 obj 目录下。**

**2，执行根目录下的mkconfig文件：**

|  |  |
| --- | --- |
| 1  2 | %\_config::  unconfig      @$(MKCONFIG) -A $(@:\_config=) |

   即：         mkconfig -A am335x\_evm

**其功能实现如下：**

设为 ： ./ （即当前目录，即uboot根目录）  
1，创建include目录：./am335x/include/  
        创建 asm目录：./am335x/include/asm/  
            创建符号链接 asm/arch，即 ./am335x/include/asm/arch 指向 ./arch/arm/include/asm/arch-ti81xx/ 目录  
            创建符号链接 asm/proc，即 ./am335x/include/asm/proc 指向 ./arch/arm/include/asm/proc-armv/ 目录.  
  
2，创建include2目录：./am335x/include2/  
        创建符号链接asm，即 ./am335x/include2/asm 指向 ./arch/arm/include/asm      
  
**./am335x/**  
**|------include/**  
**|         |------asm/**  
**|         |       |------arch**  
**|         |       |------proc**  
**|         |------config.mk**  
**|         |------config.h**  
**|------include2/**  
**|------asm**

**3，执行make：**

            运行 config.mk 文件。

**以下是uboot根目录下的makefile，mkconfig文件的详解：**

**makefile 详解：**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191  192  193  194  195  196  197  198  199  200  201  202  203  204  205  206  207  208  209  210  211  212  213  214  215  216  217  218  219  220  221  222  223  224  225  226  227  228  229  230  231  232  233  234  235  236  237  238  239  240  241  242  243  244  245  246  247  248  249  250  251  252  253  254  255  256  257  258  259  260  261  262  263  264  265  266  267  268  269  270  271  272  273  274  275  276  277  278  279  280  281  282  283  284  285  286  287  288  289  290  291  292  293  294  295  296  297  298  299  300  301  302  303  304  305  306  307  308  309  310  311  312  313  314  315  316  317  318  319  320  321  322  323  324  325  326  327  328  329  330  331  332  333  334  335  336  337  338  339  340  341  342  343  344  345  346  347  348  349  350  351  352  353  354  355  356  357  358  359  360  361  362  363  364  365  366  367  368  369  370  371  372  373  374  375  376  377  378  379  380  381  382  383  384  385  386  387  388  389  390  391  392  393  394  395  396  397  398  399  400  401  402  403  404  405  406  407  408  409  410  411  412  413  414  415  416  417  418  419  420  421  422  423  424  425  426  427  428  429  430  431  432  433  434  435  436  437  438  439  440  441  442  443  444  445  446  447  448  449  450  451  452  453  454  455  456  457  458  459  460  461  462  463  464  465  466  467  468  469  470  471  472  473  474  475  476  477  478  479  480  481  482  483  484  485  486  487  488  489  490  491  492  493  494  495  496  497  498  499  500  501  502  503  504  505  506  507  508  509  510  511  512  513  514  515  516  517  518  519  520  521  522  523  524  525  526  527  528  529  530  531  532  533  534  535  536  537  538  539  540  541  542  543  544  545  546  547  548  549  550  551  552  553  554  555  556  557  558  559  560  561  562  563  564  565  566  567  568  569  570  571  572  573  574  575  576  577  578  579  580  581  582  583  584  585  586  587  588  589  590  591  592  593  594  595  596  597  598  599  600  601  602  603  604  605  606  607  608  609  610  611  612  613  614  615  616  617  618  619  620  621  622  623  624  625  626  627  628  629  630  631  632  633  634  635  636  637  638  639  640  641  642  643  644  645  646  647  648  649  650  651  652  653  654  655  656  657  658  659  660  661  662  663  664  665  666  667  668  669  670  671  672  673  674  675  676  677  678  679  680  681  682  683  684  685  686  687  688  689  690  691  692  693  694  695  696  697  698  699  700  701  702  703  704  705  706  707  708  709  710  711  712  713  714  715  716  717  718  719  720  721  722  723  724  725  726  727  728  729  730  731  732  733  734  735  736  737  738  739  740  741  742  743  744  745  746  747  748  749  750  751  752  753  754  755  756  757  758  759  760  761  762  763  764  765  766  767  768  769  770  771  772  773  774  775  776  777  778  779  780  781  782  783  784  785  786  787  788  789  790  791  792  793  794  795  796  797  798  799  800  801  802  803  804  805  806  807  808  809  810  811  812  813  814  815  816  817  818  819  820  821  822  823  824  825  826  827  828  829  830  831  832  833  834  835  836  837  838  839  840  841  842  843  844  845  846  847  848  849  850  851  852  853  854  855  856  857  858  859  860  861  862  863  864  865  866  867  868  869  870  871  872  873  874  875  876  877  878  879  880  881  882  883  884  885  886  887  888  889  890  891  892  893  894  895  896  897  898  899  900  901  902  903  904  905  906  907  908  909  910  911  912  913  914  915  916  917  918  919  920  921  922  923  924  925  926  927  928  929  930  931  932  933  934  935  936  937  938  939  940  941  942  943  944  945  946  947  948  949  950  951  952  953  954  955  956  957  958  959  960  961  962  963  964  965  966  967  968  969  970  971  972  973  974  975  976  977  978  979  980  981  982  983  984  985  986  987  988  989  990  991  992  993  994  995  996  997  998  999  1000  1001  1002  1003  1004  1005  1006  1007  1008  1009  1010  1011  1012  1013  1014  1015  1016  1017  1018  1019  1020  1021  1022  1023  1024  1025  1026  1027  1028  1029  1030  1031  1032  1033  1034  1035  1036  1037  1038  1039  1040  1041  1042  1043  1044  1045  1046  1047  1048  1049  1050  1051  1052  1053  1054  1055  1056  1057  1058  1059  1060  1061  1062  1063  1064  1065  1066  1067  1068  1069  1070  1071  1072  1073  1074  1075  1076  1077  1078  1079  1080  1081  1082  1083  1084  1085  1086  1087  1088  1089  1090  1091  1092  1093  1094  1095  1096  1097  1098  1099  1100  1101  1102  1103  1104  1105  1106  1107  1108  1109  1110  1111  1112  1113  1114  1115  1116  1117  1118  1119  1120  1121  1122  1123  1124  1125  1126  1127  1128  1129  1130  1131  1132  1133  1134  1135  1136  1137  1138  1139  1140  1141  1142  1143  1144  1145  1146  1147  1148  1149  1150  1151  1152  1153  1154  1155  1156  1157  1158  1159  1160  1161  1162  1163  1164  1165  1166  1167  1168  1169  1170  1171  1172  1173  1174  1175  1176  1177  1178  1179  1180  1181  1182  1183  1184  1185  1186  1187  1188  1189  1190  1191  1192  1193  1194  1195  1196  1197  1198  1199  1200  1201  1202  1203  1204  1205  1206  1207  1208  1209  1210  1211  1212  1213  1214  1215  1216  1217  1218  1219  1220  1221  1222  1223  1224  1225  1226  1227  1228  1229  1230  1231  1232  1233  1234  1235  1236  1237  1238  1239  1240  1241  1242  1243  1244  1245  1246  1247  1248  1249  1250  1251  1252  1253  1254  1255  1256  1257  1258  1259  1260  1261  1262  1263  1264  1265  1266  1267  1268  1269  1270  1271  1272  1273  1274  1275  1276  1277  1278  1279  1280  1281  1282  1283  1284  1285  1286  1287  1288  1289  1290  1291  1292  1293  1294  1295  1296  1297  1298  1299  1300  1301  1302  1303  1304  1305  1306  1307  1308  1309 | #  # (C) Copyright 2000-2011  # Wolfgang Denk, DENX Software Engineering, wd@denx.de.  #  # See file CREDITS for list of people who contributed to this  # project.  #  # This program is free software; you can redistribute it and/or  # modify it under the terms of the GNU General Public License as  # published by the Free Software Foundatio; either version 2 of  # the License, or (at your option) any later version.  #  # This program is distributed in the hope that it will be useful,  # but WITHOUT ANY WARRANTY; without even the implied warranty of  # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the  # GNU General Public License for more details.  #  # You should have received a copy of the GNU General Public License  # along with this program; if not, write to the Free Software  # Foundation, Inc., 59 Temple Place, Suite 330, Boston,  # MA 02111-1307 USA  #    VERSION = 2011  PATCHLEVEL = 09  SUBLEVEL =  EXTRAVERSION =  ifneq "$(SUBLEVEL)" ""  U\_BOOT\_VERSION = $(VERSION).$(PATCHLEVEL).$(SUBLEVEL)$(EXTRAVERSION)  else  U\_BOOT\_VERSION = $(VERSION).$(PATCHLEVEL)$(EXTRAVERSION)  endif  TIMESTAMP\_FILE = $(obj)include/timestamp\_autogenerated.h  VERSION\_FILE = $(obj)include/version\_autogenerated.h    #########################################################################  #@定义主机系统架构@:  #   “sed –e”表示后面跟的是一串命令脚本，而表达式“s/abc/def/”表示要从标准输入中，  #   查找到内容为“abc”的，然后替换成“def”。其中“abc”表达式用可以使用“.”作为通配符。  #   命令“uname –m”将输出主机CPU的体系架构类型。作者的电脑使用Intel Core2系列的CPU，  #   因此“uname –m”输出“i686”。 “i686”可以匹配命令“sed -e s/i.86/i386/”中的“i.86”，  #   因此在作者的机器上执行Makefile，HOSTARCH 将被设置成“i386” 。  #########################################################################  HOSTARCH := $(shell uname -m | \      sed -e s/i.86/x86/ \          -e s/sun4u/sparc64/ \          -e s/arm.\*/arm/ \          -e s/sa110/arm/ \          -e s/ppc64/powerpc/ \          -e s/ppc/powerpc/ \          -e s/macppc/powerpc/\          -e s/sh.\*/sh/)    #########################################################################  #@定义主机操作系统类型@:  #   “uname –s”输出主机内核名字，作者使用Linux发行版Ubuntu11.10，因此“uname –s”结果是“Linux”。  #   “tr '[:upper:]' '[:lower:]'”作用是将标准输入中的所有大写字母转换为响应的小写字母。  #   因此执行结果是将HOSTOS 设置为“linux”。  #########################################################################  HOSTOS := $(shell uname -s | tr '[:upper:]' '[:lower:]' | \          sed -e 's/\(cygwin\).\*/cygwin/')    #########################################################################  #@定义执行shell脚本的shell@:  #   "$$BASH"的作用实质上是生成了字符串“$BASH”（前一个$号的作用是指明第二个$是普通的字符）。  #   若执行当前Makefile的shell中定义了“$BASH”环境变量，且文件“$BASH”是可执行文件，  #   则SHELL的值为“$BASH”。否则，若“/bin/bash”是可执行文件，则SHELL值为“/bin/bash”。  #   若以上两条都不成立，则将“sh”赋值给SHELL变量。  #   由于作者的机器安装了bash shell，SHELL 被设置为 /bin/bash。  #########################################################################  # Set shell to bash if possible, otherwise fall back to sh  SHELL := $(shell if [ -x "$$BASH" ]; then echo $$BASH; \      else if [ -x /bin/bash ]; then echo /bin/bash; \      else echo sh; fi; fi)    #导出变量HOSTARCH HOSTOS SHELL，使别的文件可以使用这些变量  export  HOSTARCH HOSTOS SHELL    # Deal with colliding definitions from tcsh etc.  VENDOR=    #因为MAKEFLAGS变量的字符串为空，找不到s字符串，所以ifeq条件为真，则XECHO = echo  # Allow for silent builds  ifeq (,$(findstring s,$(MAKEFLAGS)))  XECHO = echo  else  XECHO = :  endif    #########################################################################  #  # U-boot build supports producing a object files to the separate external  # directory. Two use cases are supported:  #  # 1) Add O= to the make command line  # 'make O=/tmp/build all'  #  # 2) Set environement variable BUILD\_DIR to point to the desired location  # 'export BUILD\_DIR=/tmp/build'  # 'make'  #  # The second approach can also be used with a MAKEALL script  # 'export BUILD\_DIR=/tmp/build'  # './MAKEALL'  #  # Command line 'O=' setting overrides BUILD\_DIR environent variable.  #  # When none of the above methods is used the local build is performed and  # the object files are placed in the source directory.  #    #########################################################################  #@设定编译输出目录@:  #   函数$( origin, variable) 输出的结果是一个字符串，输出结果由变量variable定义的方式决定，  #   若variable在命令行中定义过，则origin函数返回值为"command line"。  #   假若在命令行中执行了“export BUILD\_DIR=/tmp/build”的命令，则“$(origin O)”值为“command line”，  #   而BUILD\_DIR被设置为“/tmp/build”。  #  #   假若在命令行中执行了“make O=am335x am335x\_evm”的命令，则“$(origin O)”值为“command line”，  #   而BUILD\_DIR被设置为“am335x”。  #########################################################################  ifdef O  ifeq ("$(origin O)", "command line")  BUILD\_DIR := $(O)  endif  endif    #判断 BUILD\_DIR 变量是否为空，当前 BUILD\_DIR 为 am335x，条件为真，则 saved-output 为 am335x  ifneq ($(BUILD\_DIR),)  saved-output := $(BUILD\_DIR)    #若${BUILD\_DIR}表示的目录没有定义，则创建该目录。  # Attempt to create a output directory.  $(shell [ -d ${BUILD\_DIR} ] || mkdir -p ${BUILD\_DIR})    #若$(BUILD\_DIR)为创建失败或不存在，则将其赋值为当前目录路径（源代码目录）。  #并检查$(BUILD\_DIR)目录是否存在。  #Pwd命令用以获取当前路径  # Verify if it was successful.  BUILD\_DIR := $(shell cd $(BUILD\_DIR) && /bin/pwd)  $(if $(BUILD\_DIR),,$(error output directory "$(saved-output)" does not exist))  endif # ifneq ($(BUILD\_DIR),)    #########################################################################  #CURDIR变量指示Make当前的工作目录，由于当前Make在U-Boot顶层目录执行Makefile，  #因此CURDIR此时就是U-Boot顶层目录。  #执行完上面的代码后， SRCTREE，src变量就是U-Boot代码顶层目录，而OBJTREE，obj变量就是输出目录，  #若没有定义BUILD\_DIR环境变量，则SRCTREE，src变量与OBJTREE，obj变量都是U-Boot源代码目录。  #而MKCONFIG则表示U-Boot根目录下的mkconfig脚本。  #if函数计算OBJTREE的值，如果BUILD\_DIR不为空，if函数的值就是BUILD\_DIR，否则是CURDIR.  #CURDIR是个环境变量。代表当前文件的目录,即uboot根目录，设为 : ./。  #   CURDIR      = ./  #   OBJTREE     = ./am335x  #   SPLTREE     = ./am335x/spl  #   SRCTREE     = ./  #   TOPDIR      = ./  #   LNDIR       = ./am335x  #导出变量TOPDIR SRCTREE OBJTREE SPLTREE，使别的文件可以使用这些变量  #########################################################################  OBJTREE     := $(if $(BUILD\_DIR),$(BUILD\_DIR),$(CURDIR))  SPLTREE     := $(OBJTREE)/spl  SRCTREE     := $(CURDIR)  TOPDIR      := $(SRCTREE)  LNDIR       := $(OBJTREE)  export  TOPDIR SRCTREE OBJTREE SPLTREE    #   MKCONFIG    = ./mkconfig  #导出变量MKCONFIG  MKCONFIG    := $(SRCTREE)/mkconfig  export MKCONFIG    #########################################################################  #判断变量OBJTREE 与 SRCTREE 是否相等，OBJTREE为./am335x，SRCTREE为./变量，条件为真。  #则变量 REMOTE\_BUILD 为 1，  #导出变量 REMOTE\_BUILD  #########################################################################  ifneq ($(OBJTREE),$(SRCTREE))  REMOTE\_BUILD    := 1  export REMOTE\_BUILD  endif    #########################################################################  #判断变量OBJTREE 与 SRCTREE 是否相等，OBJTREE为./am335x，SRCTREE为./变量，条件为真。  #则  obj 为 ./am335x/  #   src 为 ./  #导出变量 obj src  #########################################################################  # $(obj) and (src) are defined in config.mk but here in main Makefile  # we also need them before config.mk is included which is the case for  # some targets like unconfig, clean, clobber, distclean, etc.  ifneq ($(OBJTREE),$(SRCTREE))  obj := $(OBJTREE)/  src := $(SRCTREE)/  else  obj :=  src :=  endif  export obj src    #失能导出变量 CDPATH的使用域，这样下文中如果定义了或使用了CDPATH变量，将不会收到导出变量CDPATH的影响。  # Make sure CDPATH settings don't interfere  unexport CDPATH    # The "tools" are needed early, so put this first  # Don't include stuff already done in $(LIBS)  SUBDIRS = tools \        examples/standalone \        examples/api    #定义SUBDIRS VERSION\_FILE伪目标  .PHONY : $(SUBDIRS) $(VERSION\_FILE)    #########################################################################  #使用“$(wildcard \*.c) ”来获取工作目录下的所有的.c 文件列表  #在当前例子中，则是为了找到 ./am335x/include/config.mk  #判断是否找到 ./am335x/include/config.mk  #########################################################################  ifeq ($(obj)include/config.mk,$(wildcard $(obj)include/config.mk))    #########################################################################  #使用“include FILENAMES... ”，make程序处理时，如果“FILENAMES ”列表  #   中的任何一个文件不能正常读取而且不存在一个创建此文件的规则时make 程序将会提示错误并退出。  #  #使用“-include FILENAMES... ”的情况是，当所包含的文件不存在或者不存在一  #   个规则去创建它，make程序会继续执行，只有真正由于不能正确完成终极目标的重建  #   时（某些必需的目标无法在当前已读取的makefile 文件内容中找到正确的重建规则），  #   才会提示致命错误并退出。  #  #为了和其它的make 程序进行兼容。也可以使用“sinclude ”来代替“-include ”（GNU所支持的方式）  #  # include/autoconf.mk(./am335x/include/autoconf.mk)文件中是与开发板相关的一些宏定义,  #   在Makefile执行过程中需要根据某些宏来确定执行哪些操作。  # include/autoconf.mk生成的规则为下文的：$(obj)include/autoconf.mk: $(obj)include/config.h  # include/autoconf.mk.dep 同理.  #########################################################################  # Include autoconf.mk before config.mk so that the config options are available  # to all top level build files.  We need the dummy all: target to prevent the  # dependency target in autoconf.mk.dep from being the default.  #执行make的时候默认就是执行make all  all:  sinclude $(obj)include/autoconf.mk.dep  sinclude $(obj)include/autoconf.mk    #包含./am335x/include/config.mk 文件,加载该文件中的ARCH CPU BOARD VENDOR SOC，并导出  # load ARCH, BOARD, and CPU configuration  include $(obj)include/config.mk  export  ARCH CPU BOARD VENDOR SOC    #########################################################################  #FOO ?= bar  #   其含义是，如果 FOO 没有被定义过，那么变量 FOO 的值就是“bar”，如果 FOO 先前被定义  #   过，那么这条语将什么也不做，其等价于：    #   ifeq ($(origin FOO), undefined)  #   FOO = bar  #   endif    #若主机架构与开发板结构相同，就使用主机的编译器，而不是交叉编译器  #当前 HOSTARCH 为 i386  #当前 ARCH     为 arm  #条件为假，则 CROSS\_COMPILE 为交叉编译器,即 CROSS\_COMPILE = arm-arago-linux-gnueabi-  #########################################################################  # set default to nothing for native builds  ifeq ($(HOSTARCH),$(ARCH))  CROSS\_COMPILE ?=  endif    #包含 ./config.mk 文件,其主要是一些变量和函数的定义,编译链接的参数设置以及依赖规则.  # load other configuration  include $(TOPDIR)/config.mk    #########################################################################  #判断是否定义了 LDSCRIPT 变量，当前并没有定义该变量，条件为真  #  #如果定义了CONFIG\_SYS\_LDSCRIPT，将CONFIG\_SYS\_LDSCRIPT代表的字符串去掉双引号后赋值给LDSCRIPT变量  #这里我们并没有定义CONFIG\_SYS\_LDSCRIPT  #########################################################################  # If board code explicitly specified LDSCRIPT or CONFIG\_SYS\_LDSCRIPT, use  # that (or fail if absent).  Otherwise, search for a linker script in a  # standard location.  ifndef LDSCRIPT      #LDSCRIPT := $(TOPDIR)/board/$(BOARDDIR)/u-boot.lds.debug      ifdef CONFIG\_SYS\_LDSCRIPT          # need to strip off double quotes          LDSCRIPT := $(subst ",,$(CONFIG\_SYS\_LDSCRIPT))      endif  endif    #BOARDDIR 定义与 ./mkconfig中，在当前例子中 BOARDDIR = $(VENDOR)/$(BOARD) = ti/am335x    #########################################################################  #如果没有用CONFIG\_SYS\_LDSCRIPT指定LDSCRIPT，那么就在以下几个地方搜  #判断是否定义了 LDSCRIPT 变量，当前并没有定义该变量，条件为真  #########################################################################  ifndef LDSCRIPT      #########################################################################      #如果CONFIG\_NAND\_U\_BOOT变量是否等于 y，当前没有定义CONFIG\_NAND\_U\_BOOT变量，条件为假。      #则不执行条件中的代码      #########################################################################      ifeq ($(CONFIG\_NAND\_U\_BOOT),y)          LDSCRIPT := $(TOPDIR)/board/$(BOARDDIR)/u-boot-nand.lds          ifeq ($(wildcard $(LDSCRIPT)),)              LDSCRIPT := $(TOPDIR)/$(CPUDIR)/u-boot-nand.lds          endif      endif        #########################################################################      #判断变量 LDSCRIPT 是否为空，当前 LDSCRIPT 为空，条件为真，执行条件中的代码。      # BOARDDIR 定义与 ./mkconfig中，在当前例子中 BOARDDIR = $(VENDOR)/$(BOARD) = ti/am335x      # TOPDIR = ./      # LDSCRIPT := $(TOPDIR)/board/$(BOARDDIR)/u-boot.lds -> LDSCRIPT = ./board/ti/am335x/u-boot.lds,      # 查找 ./board/ti/am335x/ 目录下的 u-boot.lds 文件，没有找到对应的文件，所以  LDSCRIPT 为空。      #########################################################################      ifeq ($(wildcard $(LDSCRIPT)),)          LDSCRIPT := $(TOPDIR)/board/$(BOARDDIR)/u-boot.lds      endif        #########################################################################      #判断变量 LDSCRIPT 是否为空，当前 LDSCRIPT 为空，条件为真，执行条件中的代码。      # 在顶层 config.mk 中 得 CPUDIR  = CPUDIR=arch/$(ARCH)/cpu/$(CPU) = arch/arm/cpu/armv7      # TOPDIR = ./      # LDSCRIPT := $(TOPDIR)/$(CPUDIR)/u-boot.lds -> LDSCRIPT = ./arch/arm/cpu/armv7/u-boot.lds,      # 查找 ./arch/arm/cpu/armv7/ 目录下的 u-boot.lds 文件，找到了对应的 u-boot.lds 文件，      # 因此，LDSCRIPT = ./arch/arm/cpu/armv7/u-boot.lds      #########################################################################      ifeq ($(wildcard $(LDSCRIPT)),)          LDSCRIPT := $(TOPDIR)/$(CPUDIR)/u-boot.lds      endif        #########################################################################      #判断变量 LDSCRIPT 是否为空， 当前 LDSCRIPT = ./arch/arm/cpu/armv7/u-boot.lds，      #条件为假，不执行条件中的代码      #########################################################################      ifeq ($(wildcard $(LDSCRIPT)),)  $(error could not find linker script)      endif  endif    #########################################################################  # U-Boot objects....order is important (i.e. start must be first)  # U-boot需要的目标文件，顺序很重要，start.o必须放第一位  # 在顶层 config.mk 中 得 CPUDIR  = CPUDIR=arch/$(ARCH)/cpu/$(CPU) = arch/arm/cpu/armv7  OBJS  = $(CPUDIR)/start.o    # 根据 ./am335x/include/config.mk 得 CPU    = armv7  # 在顶层 config.mk 中 得 CPUDIR  = CPUDIR=arch/$(ARCH)/cpu/$(CPU) = arch/arm/cpu/armv7  ifeq ($(CPU),x86)  OBJS += $(CPUDIR)/start16.o  OBJS += $(CPUDIR)/resetvec.o  endif  ifeq ($(CPU),ppc4xx)  OBJS += $(CPUDIR)/resetvec.o  endif  ifeq ($(CPU),mpc85xx)  OBJS += $(CPUDIR)/resetvec.o  endif    # obj = ./am335x  OBJS := $(addprefix $(obj),$(OBJS))    # 根据 ./am335x/include/config.mk 得 VENDOR = ti  LIBS  = lib/libgeneric.o  LIBS += lib/lzma/liblzma.o  LIBS += lib/lzo/liblzo.o  LIBS += lib/zlib/libz.o  LIBS += $(shell if [ -f board/$(VENDOR)/common/Makefile ]; then echo \      "board/$(VENDOR)/common/lib$(VENDOR).o"; fi)  LIBS += $(CPUDIR)/lib$(CPU).o    #根据 ./am335x/include/config.mk 得 SOC    = ti81xx  ifdef SOC  LIBS += $(CPUDIR)/$(SOC)/lib$(SOC).o  endif  ifeq ($(CPU),ixp)  LIBS += arch/arm/cpu/ixp/npe/libnpe.o  endif    # 根据 ./am335x/include/config.mk 得 ARCH   = arm  LIBS += arch/$(ARCH)/lib/lib$(ARCH).o  LIBS += fs/cramfs/libcramfs.o fs/fat/libfat.o fs/fdos/libfdos.o fs/jffs2/libjffs2.o \      fs/reiserfs/libreiserfs.o fs/ext2/libext2fs.o fs/yaffs2/libyaffs2.o \      fs/ubifs/libubifs.o  LIBS += net/libnet.o  LIBS += disk/libdisk.o  LIBS += drivers/bios\_emulator/libatibiosemu.o  LIBS += drivers/block/libblock.o  LIBS += drivers/dma/libdma.o  LIBS += drivers/fpga/libfpga.o  LIBS += drivers/gpio/libgpio.o  LIBS += drivers/hwmon/libhwmon.o  LIBS += drivers/i2c/libi2c.o  LIBS += drivers/input/libinput.o  LIBS += drivers/misc/libmisc.o  LIBS += drivers/mmc/libmmc.o  LIBS += drivers/mtd/libmtd.o  LIBS += drivers/mtd/nand/libnand.o  LIBS += drivers/mtd/onenand/libonenand.o  LIBS += drivers/mtd/ubi/libubi.o  LIBS += drivers/mtd/spi/libspi\_flash.o  LIBS += drivers/net/libnet.o  LIBS += drivers/net/phy/libphy.o  LIBS += drivers/pci/libpci.o  LIBS += drivers/pcmcia/libpcmcia.o  LIBS += drivers/power/libpower.o  LIBS += drivers/spi/libspi.o  ifeq ($(CPU),mpc83xx)  LIBS += drivers/qe/libqe.o  LIBS += arch/powerpc/cpu/mpc8xxx/lib8xxx.o  endif  ifeq ($(CPU),mpc85xx)  LIBS += drivers/qe/libqe.o  LIBS += arch/powerpc/cpu/mpc8xxx/ddr/libddr.o  LIBS += arch/powerpc/cpu/mpc8xxx/lib8xxx.o  endif  ifeq ($(CPU),mpc86xx)  LIBS += arch/powerpc/cpu/mpc8xxx/ddr/libddr.o  LIBS += arch/powerpc/cpu/mpc8xxx/lib8xxx.o  endif  LIBS += drivers/rtc/librtc.o  LIBS += drivers/serial/libserial.o  LIBS += drivers/twserial/libtws.o  LIBS += drivers/usb/eth/libusb\_eth.o  LIBS += drivers/usb/gadget/libusb\_gadget.o  LIBS += drivers/usb/host/libusb\_host.o  LIBS += drivers/usb/musb/libusb\_musb.o  LIBS += drivers/usb/phy/libusb\_phy.o  LIBS += drivers/video/libvideo.o  LIBS += drivers/watchdog/libwatchdog.o  LIBS += common/libcommon.o  LIBS += lib/libfdt/libfdt.o  LIBS += api/libapi.o  LIBS += post/libpost.o    #根据 ./am335x/include/config.mk 得 SOC    = ti81xx  ifeq ($(SOC),ti81xx)  LIBS += $(CPUDIR)/omap-common/libomap-common.o  endif  ifeq ($(SOC),omap3)  LIBS += $(CPUDIR)/omap-common/libomap-common.o  endif  ifeq ($(SOC),omap4)  LIBS += $(CPUDIR)/omap-common/libomap-common.o  endif    ifeq ($(SOC),s5pc1xx)  LIBS += $(CPUDIR)/s5p-common/libs5p-common.o  endif  ifeq ($(SOC),s5pc2xx)  LIBS += $(CPUDIR)/s5p-common/libs5p-common.o  endif    #########################################################################/\*  #   8.3.6 $(addprefix PREFIX,NAMES…)  #   函数名称：加前缀函数—addprefix。  #  #   函数功能：为“NAMES…”中的每一个文件名添加前缀“PREFIX”。参数“ NAMES…”  #   是空格分割的文件名序列，将“SUFFIX”添加到此序列的每一个文件名之前。  #  #   返回值：以单空格分割的添加了前缀“PREFIX”的文件名序列。  #  #函数说明：  #示例：  #$(addprefix src/,foo bar)  #返回值为“src/foo src/bar”。  #  #  # obj = ./am335x  #########################################################################\*/  LIBS := $(addprefix $(obj),$(sort $(LIBS)))  #定义LIBS TIMESTAMP\_FILE伪目标  #TIMESTAMP\_FILE = $(obj)include/timestamp\_autogenerated.h -> ./am335x/include/timestamp\_autogenerated.h  .PHONY : $(LIBS) $(TIMESTAMP\_FILE)    # BOARDDIR 定义与 ./mkconfig中，在当前例子中 BOARDDIR = $(VENDOR)/$(BOARD) = ti/am335x  #根据 ./am335x/include/config.mk 得 BOARD  = am335x  LIBBOARD = board/$(BOARDDIR)/lib$(BOARD).o  LIBBOARD := $(addprefix $(obj),$(LIBBOARD))    # 没有定义 USE\_PRIVATE\_LIBGCC  # Add GCC lib  ifdef USE\_PRIVATE\_LIBGCC  ifeq ("$(USE\_PRIVATE\_LIBGCC)", "yes")  PLATFORM\_LIBGCC = $(OBJTREE)/arch/$(ARCH)/lib/libgcc.o  else  PLATFORM\_LIBGCC = -L $(USE\_PRIVATE\_LIBGCC) -lgcc  endif  else  # 在顶层 config.mk(./config.mk) 中 得CC,CFLAGS:  #CC = $(CROSS\_COMPILE)gcc  #CROSS\_COMPILE = arm-arago-linux-gnueabi-  #CC = arm-arago-linux-gnueabi-gcc  PLATFORM\_LIBGCC = -L $(shell dirname `$(CC) $(CFLAGS) -print-libgcc-file-name`) -lgcc  endif  PLATFORM\_LIBS += $(PLATFORM\_LIBGCC)  #导出变量 PLATFORM\_LIBS  export PLATFORM\_LIBS    # Special flags for CPP when processing the linker script.  # Pass the version down so we can handle backwards compatibility  # on the fly.  LDPPFLAGS += \      -include $(TOPDIR)/include/u-boot/u-boot.lds.h \      $(shell $(LD) --version | \        sed -ne 's/GNU ld version \([0-9][0-9]\*\)\.\([0-9][0-9]\*\).\*/-DLD\_MAJOR=\1 -DLD\_MINOR=\2/p')    \_\_OBJS := $(subst $(obj),,$(OBJS))  \_\_LIBS := $(subst $(obj),,$(LIBS)) $(subst $(obj),,$(LIBBOARD))    #########################################################################  #########################################################################  #没有定义变量 CONFIG\_BOARD\_SIZE\_LIMIT，因此 BOARD\_SIZE\_CHECK 为空  ifneq ($(CONFIG\_BOARD\_SIZE\_LIMIT),)  BOARD\_SIZE\_CHECK = \      @actual=`wc -c $@ | awk '{print $$1}'`; \      limit=$(CONFIG\_BOARD\_SIZE\_LIMIT); \      if test $$actual -gt $$limit; then \          echo "$@ exceeds file size limit:"; \          echo "  limit:  $$limit bytes"; \          echo "  actual: $$actual bytes"; \          echo "  excess: $$((actual - limit)) bytes"; \          exit 1; \      fi  else  BOARD\_SIZE\_CHECK =  endif    #最终生成的各种镜像文件，及其生成方法  # Always append ALL so that arch config.mk's can add custom ones  ALL-y += $(obj)u-boot.srec $(obj)u-boot.bin $(obj)System.map    ALL-$(CONFIG\_NAND\_U\_BOOT) += $(obj)u-boot-nand.bin  ALL-$(CONFIG\_ONENAND\_U\_BOOT) += $(obj)u-boot-onenand.bin  ONENAND\_BIN ?= $(obj)onenand\_ipl/onenand-ipl-2k.bin  ALL-$(CONFIG\_MMC\_U\_BOOT) += $(obj)mmc\_spl/u-boot-mmc-spl.bin  ALL-$(CONFIG\_SPL) += $(obj)spl/u-boot-spl.bin    all:        $(ALL-y)    # 在uboot顶层 config.mk 中 得 OBJCOPY = $(CROSS\_COMPILE)objcopy -> arm-arago-linux-gnueabi-objcopy,  # 生成  .hex 格式的文件  $(obj)u-boot.hex:   $(obj)u-boot          $(OBJCOPY) ${OBJCFLAGS} -O ihex $< $@    # 在uboot顶层 config.mk 中 得 OBJCOPY = $(CROSS\_COMPILE)objcopy -> arm-arago-linux-gnueabi-objcopy,  # 生成某个格式的文件  $(obj)u-boot.srec:  $(obj)u-boot          $(OBJCOPY) -O srec $< $@    # BOARD\_SIZE\_CHECK 为空  # 生成 .bin 文件  $(obj)u-boot.bin:   $(obj)u-boot          $(OBJCOPY) ${OBJCFLAGS} -O binary $< $@          $(BOARD\_SIZE\_CHECK)    $(obj)u-boot.ldr:   $(obj)u-boot          $(CREATE\_LDR\_ENV)          $(LDR) -T $(CONFIG\_BFIN\_CPU) -c $@ $< $(LDR\_FLAGS)          $(BOARD\_SIZE\_CHECK)    $(obj)u-boot.ldr.hex:   $(obj)u-boot.ldr          $(OBJCOPY) ${OBJCFLAGS} -O ihex $< $@ -I binary    $(obj)u-boot.ldr.srec:  $(obj)u-boot.ldr          $(OBJCOPY) ${OBJCFLAGS} -O srec $< $@ -I binary    # 在uboot顶层 config.mk 中 得 CONFIG\_SYS\_TEXT\_BASE=0x80800000  $(obj)u-boot.img:   $(obj)u-boot.bin          $(obj)tools/mkimage -A $(ARCH) -T firmware -C none \          -O u-boot -a $(CONFIG\_SYS\_TEXT\_BASE) -e 0 \          -n $(shell sed -n -e 's/.\*U\_BOOT\_VERSION//p' $(VERSION\_FILE) | \              sed -e 's/"[     ]\*$$/ for $(BOARD) board"/') \          -d $< $@    $(obj)u-boot.imx:       $(obj)u-boot.bin          $(obj)tools/mkimage -n  $(CONFIG\_IMX\_CONFIG) -T imximage \          -e $(CONFIG\_SYS\_TEXT\_BASE) -d $< $@    $(obj)u-boot.kwb:       $(obj)u-boot.bin          $(obj)tools/mkimage -n $(CONFIG\_SYS\_KWD\_CONFIG) -T kwbimage \          -a $(CONFIG\_SYS\_TEXT\_BASE) -e $(CONFIG\_SYS\_TEXT\_BASE) -d $< $@    $(obj)u-boot.sha1:  $(obj)u-boot.bin          $(obj)tools/ubsha1 $(obj)u-boot.bin    $(obj)u-boot.dis:   $(obj)u-boot          $(OBJDUMP) -d $< > $@    $(obj)u-boot.ubl:       $(obj)u-boot-nand.bin          $(obj)tools/mkimage -n $(UBL\_CONFIG) -T ublimage \          -e $(CONFIG\_SYS\_TEXT\_BASE) -d $< $@    GEN\_UBOOT = \          UNDEF\_SYM=`$(OBJDUMP) -x $(LIBBOARD) $(LIBS) | \          sed  -n -e 's/.\*\($(SYM\_PREFIX)\_\_u\_boot\_cmd\_.\*\)/-u\1/p'|sort|uniq`;\          cd $(LNDIR) && $(LD) $(LDFLAGS) $(LDFLAGS\_$(@F)) $$UNDEF\_SYM $(\_\_OBJS) \              --start-group $(\_\_LIBS) --end-group $(PLATFORM\_LIBS) \              -Map u-boot.map -o u-boot    #########################################################################/\*  #u-boot ELF文件镜像的生成是最关键的.  #u-boot 依赖depend $(SUBDIR\_TOOLS) $(OBJS) $(LIBBOARD) $(LIBS) $(LDSCRIPT) $(obj)u-boot.lds.  #然后用$(GEN\_UBOOT)生成最后的u-boot,GEN\_UBOOT就是用 ld 链接的过程  #########################################################################\*/  $(obj)u-boot:   depend \          $(SUBDIRS) $(OBJS) $(LIBBOARD) $(LIBS) $(LDSCRIPT) $(obj)u-boot.lds          $(GEN\_UBOOT)    #没有定义 CONFIG\_KALLSYMS  ifeq ($(CONFIG\_KALLSYMS),y)          smap=`$(call SYSTEM\_MAP,u-boot) | \              awk '$$2 ~ /[tTwW]/ {printf $$1 $$3 "\\\\000"}'` ; \          $(CC) $(CFLAGS) -DSYSTEM\_MAP="\"$${smap}\"" \              -c common/system\_map.c -o $(obj)common/system\_map.o          $(GEN\_UBOOT) $(obj)common/system\_map.o  endif    # 在顶层 config.mk 中 得 CPUDIR  = CPUDIR=arch/$(ARCH)/cpu/$(CPU) = arch/arm/cpu/armv7  #依赖目标$(OBJS)，执行arch/arm/cpu/armv7 目录下的 makefile， 生成 ./am335x/arch/arm/cpu/armv7/start.o  #REMOTE\_BUILD := 1  #看下$(if $(REMOTE\_BUILD),$@,$(notdir $@))  #因为$(REMOTE\_BUILD)为1,所以返回的是 $@ 的值,即 OBJS;  $(OBJS):    depend          $(MAKE) -C $(CPUDIR) $(if $(REMOTE\_BUILD),$@,$(notdir $@))    #依赖目标$(LIBS)，进入到LIBS包含的很多目录，执行make,生成很多.a文件  $(LIBS):    depend $(SUBDIRS)          $(MAKE) -C $(dir $(subst $(obj),,$@))    # BOARDDIR 定义与 ./mkconfig中，在当前例子中   BOARDDIR = $(VENDOR)/$(BOARD) = ti/am335x  # 根据 ./am335x/include/config.mk 得           BOARD  = am335x  #   LIBBOARD = board/$(BOARDDIR)/lib$(BOARD).o  #   LIBBOARD := $(addprefix $(obj),$(LIBBOARD))  #   LIBBOARD = ./am335x/board/ti/am335x/libam335x.o  $(LIBBOARD):    depend $(LIBS)          $(MAKE) -C $(dir $(subst $(obj),,$@))    #伪目标SUBDIRS: 执行tools ,examples ,post,post\cpu 子目录下面的make文件  $(SUBDIRS): depend          $(MAKE) -C $@ all    #LDSCRIPT = ./arch/arm/cpu/armv7/u-boot.lds  #   其实就是把start.o和各个子目录makefile生成的库文件按照LDFLAGS连接在一起，  #   生成ELF文件u-boot 和连接时内存分配图文件u-boot.map。  $(LDSCRIPT):    depend          $(MAKE) -C $(dir $@) $(notdir $@)    $(obj)u-boot.lds: $(LDSCRIPT)          $(CPP) $(CPPFLAGS) $(LDPPFLAGS) -ansi -D\_\_ASSEMBLY\_\_ -P - <$^ >$@    nand\_spl:   $(TIMESTAMP\_FILE) $(VERSION\_FILE) depend          $(MAKE) -C nand\_spl/board/$(BOARDDIR) all    $(obj)u-boot-nand.bin:  nand\_spl $(obj)u-boot.bin          cat $(obj)nand\_spl/u-boot-spl-16k.bin $(obj)u-boot.bin > $(obj)u-boot-nand.bin    onenand\_ipl:    $(TIMESTAMP\_FILE) $(VERSION\_FILE) $(obj)include/autoconf.mk          $(MAKE) -C onenand\_ipl/board/$(BOARDDIR) all    $(obj)u-boot-onenand.bin:   onenand\_ipl $(obj)u-boot.bin          cat $(ONENAND\_BIN) $(obj)u-boot.bin > $(obj)u-boot-onenand.bin    mmc\_spl:    $(TIMESTAMP\_FILE) $(VERSION\_FILE) depend          $(MAKE) -C mmc\_spl/board/$(BOARDDIR) all    $(obj)mmc\_spl/u-boot-mmc-spl.bin:   mmc\_spl    $(obj)spl/u-boot-spl.bin:       depend          $(MAKE) -C spl all    $(TIMESTAMP\_FILE):          @LC\_ALL=C date +'#define U\_BOOT\_DATE "%b %d %C%y"' > $@          @LC\_ALL=C date +'#define U\_BOOT\_TIME "%T"' >> $@    updater:          $(MAKE) -C tools/updater all    #########################################################################/\*  #TIMESTAMP\_FILE = $(obj)include/timestamp\_autogenerated.h   -> ./am335x/include/timestamp\_autogenerated.h  #VERSION\_FILE   = $(obj)include/version\_autogenerated.h     -> ./am335x/include/version\_autogenerated.h  #obj            = ./am335x/  #SUBDIRS        = tools examples/standalone examples/api  #CPUDIR         = CPUDIR = arch/$(ARCH)/cpu/$(CPU)          -> arch/arm/cpu/armv7  #LDSCRIPT       = $(TOPDIR)/$(CPUDIR)/u-boot.lds            -> ./arch/arm/cpu/armv7/u-boot.lds  #   依赖目标depend :生成各个子目录的.depend文件，.depend列出每个目标文件的依赖文件。  #   生成方法，调用每个子目录的   make \_depend  #   对$(SUBDIRS) $(CPUDIR) $(LDSCRIPT\_MAKEFILE\_DIR)目录生成depend依赖文件;  #   而\_depend是在uboot根目录下的rules.mk（./rules.mk）中定义的，利用CC的-M选项生成依赖文件.  # 注意： 所有的 .depend 文件都是隐藏文件。  #       当前所有的 .depend 文件都生成在 ./am335x 目录下的各个子目录中。  #########################################################################\*/  # Explicitly make \_depend in subdirs containing multiple targets to prevent  # parallel sub-makes creating .depend files simultaneously.  depend dep: $(TIMESTAMP\_FILE) $(VERSION\_FILE) \          $(obj)include/autoconf.mk \          $(obj)include/generated/generic-asm-offsets.h \          $(obj)include/generated/asm-offsets.h          for dir in $(SUBDIRS) $(CPUDIR) $(dir $(LDSCRIPT)) ; do \              $(MAKE) -C $$dir \_depend ; done    TAG\_SUBDIRS = $(SUBDIRS)  TAG\_SUBDIRS += $(dir $(\_\_LIBS))  TAG\_SUBDIRS += include    FIND := find  FINDFLAGS := -L    tags ctags:          ctags -w -o $(obj)ctags `$(FIND) $(FINDFLAGS) $(TAG\_SUBDIRS) \                          -name '\*.[chS]' -print`    etags:          etags -a -o $(obj)etags `$(FIND) $(FINDFLAGS) $(TAG\_SUBDIRS) \                          -name '\*.[chS]' -print`  cscope:          $(FIND) $(FINDFLAGS) $(TAG\_SUBDIRS) -name '\*.[chS]' -print > \                          cscope.files          cscope -b -q -k    SYSTEM\_MAP = \          $(NM) $1 | \          grep -v '\(compiled\)\|\(\.o$$\)\|\( [aUw] \)\|\(\.\.ng$$\)\|\(LASH[RL]DI\)' | \          LC\_ALL=C sort  $(obj)System.map:   $(obj)u-boot          @$(call SYSTEM\_MAP,$<) > $(obj)System.map    #  # Auto-generate the autoconf.mk file (which is included by all makefiles)  #  # This target actually generates 2 files; autoconf.mk and autoconf.mk.dep.  # the dep file is only include in this top level makefile to determine when  # to regenerate the autoconf.mk file.  $(obj)include/autoconf.mk.dep: $(obj)include/config.h include/common.h      @$(XECHO) Generating $@ ; \      set -e ; \      : Generate the dependancies ; \      $(CC) -x c -DDO\_DEPS\_ONLY -M $(HOSTCFLAGS) $(CPPFLAGS) \          -MQ $(obj)include/autoconf.mk include/common.h > $@    #########################################################################/\*  #   include/autoconf.mk依赖于make <board\_name>\_config 命令生成的include/config.h。  #       因此执行make <board\_name>\_config命令后再执行make all将更新include/autoconf.mk。  #  #   编译选项“-dM”的作用是输出include/common.h中定义的所有宏。  #       根据上面的规则，编译器提取include/common.h中定义的宏，  #       然后输出给tools/scripts/define2mk.sed脚本处理，处理的结果就是include/autoconf.mk文件。  #       其中tools/scripts/define2mk.sed脚本的主要完成了在include/common.h中查找和处理以“CONFIG\_”开头的宏定义的功能。  #  #   include/common.h文件包含了include/config.h（./am335x/include/config.h）文件，  #       而include/config.h文件又包含了以下4 个文件：  #           #include <config\_cmd\_defaults.h>  位于： ./include/config\_cmd\_defaults.h  #           #include <config\_defaults.h>      位于： ./include/config\_defaults.h  #           #include <configs/am335x\_evm.h>       位于： ./include/configs/am335x\_evm.h  #           #include <asm/config.h>               位于： ./arch/arm/include/asm/config.h  # 。因此include/autoconf.mk实质上就是以上5个文件中“CONFIG\_”开头的有效的宏定义的集合。  #########################################################################\*/  $(obj)include/autoconf.mk: $(obj)include/config.h      @$(XECHO) Generating $@ ; \      set -e ; \      : Extract the config macros ; \      $(CPP) $(CFLAGS) -DDO\_DEPS\_ONLY -dM include/common.h | \          sed -n -f tools/scripts/define2mk.sed > $@.tmp && \      mv $@.tmp $@    $(obj)include/generated/generic-asm-offsets.h:  $(obj)include/autoconf.mk.dep \      $(obj)lib/asm-offsets.s      @$(XECHO) Generating $@      tools/scripts/make-asm-offsets $(obj)lib/asm-offsets.s $@    $(obj)lib/asm-offsets.s:    $(obj)include/autoconf.mk.dep \      $(src)lib/asm-offsets.c      @mkdir -p $(obj)lib      $(CC) -DDO\_DEPS\_ONLY \          $(CFLAGS) $(CFLAGS\_$(BCURDIR)/$(@F)) $(CFLAGS\_$(BCURDIR)) \          -o $@ $(src)lib/asm-offsets.c -c -S    $(obj)include/generated/asm-offsets.h:  $(obj)include/autoconf.mk.dep \      $(obj)$(CPUDIR)/$(SOC)/asm-offsets.s      @echo Generating $@      tools/scripts/make-asm-offsets $(obj)$(CPUDIR)/$(SOC)/asm-offsets.s $@    $(obj)$(CPUDIR)/$(SOC)/asm-offsets.s:   $(obj)include/autoconf.mk.dep      @mkdir -p $(obj)$(CPUDIR)/$(SOC)      if [ -f $(src)$(CPUDIR)/$(SOC)/asm-offsets.c ];then \          $(CC) -DDO\_DEPS\_ONLY \          $(CFLAGS) $(CFLAGS\_$(BCURDIR)/$(@F)) $(CFLAGS\_$(BCURDIR)) \              -o $@ $(src)$(CPUDIR)/$(SOC)/asm-offsets.c -c -S; \      else \          touch $@; \      fi    #########################################################################  else    # !config.mk ; ifeq ($(obj)include/config.mk,$(wildcard $(obj)include/config.mk))  all $(obj)u-boot.hex $(obj)u-boot.srec $(obj)u-boot.bin \  $(obj)u-boot.img $(obj)u-boot.dis $(obj)u-boot \  $(filter-out tools,$(SUBDIRS)) $(TIMESTAMP\_FILE) \  updater depend dep tags ctags etags cscope $(obj)System.map:      @echo "System not configured - see README" >&2      @ exit 1    tools: $(VERSION\_FILE)      $(MAKE) -C $@ all  endif   # config.mk ; ifeq ($(obj)include/config.mk,$(wildcard $(obj)include/config.mk))    #生成版本信息到版本文件VERSION\_FILE中  $(VERSION\_FILE):          @mkdir -p $(dir $(VERSION\_FILE))          @( localvers='$(shell $(TOPDIR)/tools/setlocalversion $(TOPDIR))' ; \             printf '#define PLAIN\_VERSION "%s%s"\n' \              "$(U\_BOOT\_VERSION)" "$${localvers}" ; \             printf '#define U\_BOOT\_VERSION "U-Boot %s%s"\n' \              "$(U\_BOOT\_VERSION)" "$${localvers}" ; \          ) > $@.tmp          @( printf '#define CC\_VERSION\_STRING "%s"\n' \           '$(shell $(CC) --version | head -n 1)' )>>  $@.tmp          @( printf '#define LD\_VERSION\_STRING "%s"\n' \           '$(shell $(LD) -v | head -n 1)' )>>  $@.tmp          @cmp -s $@ $@.tmp && rm -f $@.tmp || mv -f $@.tmp $@    easylogo env gdb:      $(MAKE) -C tools/$@ all MTD\_VERSION=${MTD\_VERSION}  gdbtools: gdb    tools-all: easylogo env gdb $(VERSION\_FILE)      $(MAKE) -C tools HOST\_TOOLS\_ALL=y    .PHONY : CHANGELOG  CHANGELOG:      git log --no-merges U-Boot-1\_1\_5.. | \      unexpand -a | sed -e 's/\s\s\*$$//' > $@    include/license.h: tools/bin2header COPYING      cat COPYING | gzip -9 -c | ./tools/bin2header license\_gzip > include/license.h  #########################################################################    unconfig:      @rm -f $(obj)include/config.h $(obj)include/config.mk \          $(obj)board/\*/config.tmp $(obj)board/\*/\*/config.tmp \          $(obj)include/autoconf.mk $(obj)include/autoconf.mk.dep    #########################################################################  #   %代表着任意字符  #%\_config目标后面是双冒号，而我们平常看的只有一个冒号，这个就是makefile 的双冒号规则了，  #而平常我们见的单冒号就是普通规则。Makefile 中规定：一个目标可以出现在多个规则中。  #但是这些规则必须是同一类型的规则，要么都是普通规则，要么都是双冒号规则。  #而不允许一个目标同时出现在两种不同类型的规则中。双冒号规则和普通规则的处理的不同点表现在以下几个方面：  #1. 双冒号规则中，当依赖文件比目标更新时。规则将会被执行。对于一个没有依赖而只有命令行的双冒号规则，  #   当引用此目标时，规则的命令将会被无条件执行。而普通规则，当规则的目标文件存在时，  #   此规则的命令永远不会被执行（目标文件永远是最新的）。  #2. 当同一个文件作为多个双冒号规则的目标时。这些不同的规则会被独立的处理，  #   而不是像普通规则那样合并所有的依赖到一个目标文件。这就意味着对这些规则的处理就像多个不同  #   的普通规则一样。就是说多个双冒号规则中的每一个的依赖文件被改变之后，make只执行此规则定义的命令，  #   而其它的以这个文件作为目标的双冒号规则将不会被执行。    #@的作用是在执行这条命令的时候不进行显示，$(MKCONFIG)是取变量MKCONFIG，  #由MKCONFIG := $(SRCTREE)/mkconfig这条语句知，就是当前目录下的mkconfig文件，  #$(@:\_config=)的意思是，讲目标文件名字中含有的\_config用等号后面的的字符替换掉，  #这里＝后面为空，所以其效果就是把\_config去掉  # 格式为“$(VAR:A=B)”(或者“${VAR:A=B}”),意思是:替换变量“VAR”中所有“A”字符结尾的字为“B”结尾的字。  #########################################################################  %\_config::  unconfig      @$(MKCONFIG) -A $(@:\_config=)    #########################################################################  #读取uboot根目录下的 boards.cfg 文件生成 .boards.depend 隐藏文件，该文件位于 obj 目录下。  #########################################################################  sinclude $(obj).boards.depend  $(obj).boards.depend:   boards.cfg      awk '(NF && $$1 !~ /^#/) { print $$1 ": " $$1 "\_config; $$(MAKE)" }' $< > $@    #  # Functions to generate common board directory names  #  lcname  = $(shell echo $(1) | sed -e 's/\(.\*\)\_config/\L\1/')  ucname  = $(shell echo $(1) | sed -e 's/\(.\*\)\_config/\U\1/')    #########################################################################  ## Coldfire  #########################################################################    astro\_mcf5373l\_config \  astro\_mcf5373l\_RAM\_config : unconfig      @$(MKCONFIG) -n $@ -t $@ astro\_mcf5373l m68k mcf532x mcf5373l astro    M52277EVB\_config \  M52277EVB\_spansion\_config \  M52277EVB\_stmicro\_config :  unconfig      @case "$@" in \      M52277EVB\_config)       FLASH=SPANSION;; \      M52277EVB\_spansion\_config)  FLASH=SPANSION;; \      M52277EVB\_stmicro\_config)   FLASH=STMICRO;; \      esac; \      if [ "$${FLASH}" = "SPANSION" ] ; then \          echo "#define CONFIG\_SYS\_SPANSION\_BOOT" >> $(obj)include/config.h ; \          echo "CONFIG\_SYS\_TEXT\_BASE = 0x00000000" > $(obj)board/freescale/m52277evb/config.tmp ; \          cp $(obj)board/freescale/m52277evb/u-boot.spa $(obj)board/freescale/m52277evb/u-boot.lds ; \      fi; \      if [ "$${FLASH}" = "STMICRO" ] ; then \          echo "#define CONFIG\_CF\_SBF"    >> $(obj)include/config.h ; \          echo "#define CONFIG\_SYS\_STMICRO\_BOOT"  >> $(obj)include/config.h ; \          echo "CONFIG\_SYS\_TEXT\_BASE = 0x43E00000" > $(obj)board/freescale/m52277evb/config.tmp ; \          cp $(obj)board/freescale/m52277evb/u-boot.stm $(obj)board/freescale/m52277evb/u-boot.lds ; \      fi      @$(MKCONFIG) -n $@ -a M52277EVB m68k mcf5227x m52277evb freescale    M5235EVB\_config \  M5235EVB\_Flash16\_config \  M5235EVB\_Flash32\_config:    unconfig      @case "$@" in \      M5235EVB\_config)        FLASH=16;; \      M5235EVB\_Flash16\_config)    FLASH=16;; \      M5235EVB\_Flash32\_config)    FLASH=32;; \      esac; \      if [ "$${FLASH}" != "16" ] ; then \          echo "#define NORFLASH\_PS32BIT  1" >> $(obj)include/config.h ; \          echo "CONFIG\_SYS\_TEXT\_BASE = 0xFFC00000" > $(obj)board/freescale/m5235evb/config.tmp ; \          cp $(obj)board/freescale/m5235evb/u-boot.32 $(obj)board/freescale/m5235evb/u-boot.lds ; \      else \          echo "CONFIG\_SYS\_TEXT\_BASE = 0xFFE00000" > $(obj)board/freescale/m5235evb/config.tmp ; \          cp $(obj)board/freescale/m5235evb/u-boot.16 $(obj)board/freescale/m5235evb/u-boot.lds ; \      fi      @$(MKCONFIG) -n $@ -a M5235EVB m68k mcf523x m5235evb freescale    cobra5272\_config :      unconfig      @$(MKCONFIG) $@ m68k mcf52x2 cobra5272    EB+MCF-EV123\_config :       unconfig      @mkdir -p $(obj)include      @mkdir -p $(obj)board/BuS/EB+MCF-EV123      @echo "CONFIG\_SYS\_TEXT\_BASE = 0xFFE00000"|tee $(obj)board/BuS/EB+MCF-EV123/textbase.mk      @$(MKCONFIG) -n $@ EB+MCF-EV123 m68k mcf52x2 EB+MCF-EV123 BuS    EB+MCF-EV123\_internal\_config :  unconfig      @mkdir -p $(obj)include      @mkdir -p $(obj)board/BuS/EB+MCF-EV123      @echo "CONFIG\_SYS\_TEXT\_BASE = 0xF0000000"|tee $(obj)board/BuS/EB+MCF-EV123/textbase.mk      @$(MKCONFIG) -n $@ EB+MCF-EV123 m68k mcf52x2 EB+MCF-EV123 BuS    M5329AFEE\_config \  M5329BFEE\_config :  unconfig      @case "$@" in \      M5329AFEE\_config)   NAND=0;; \      M5329BFEE\_config)   NAND=16;; \      esac; \      if [ "$${NAND}" != "0" ] ; then \          echo "#define NANDFLASH\_SIZE    $${NAND}" > $(obj)include/config.h ; \      fi      @$(MKCONFIG) -n $@ -a M5329EVB m68k mcf532x m5329evb freescale    M5373EVB\_config :   unconfig      @case "$@" in \      M5373EVB\_config)    NAND=16;; \      esac; \      if [ "$${NAND}" != "0" ] ; then \          echo "#define NANDFLASH\_SIZE    $${NAND}" > $(obj)include/config.h ; \      fi      @$(MKCONFIG) -a M5373EVB m68k mcf532x m5373evb freescale    M54451EVB\_config \  M54451EVB\_stmicro\_config :  unconfig      @case "$@" in \      M54451EVB\_config)       FLASH=NOR;; \      M54451EVB\_stmicro\_config)   FLASH=STMICRO;; \      esac; \      if [ "$${FLASH}" = "NOR" ] ; then \          echo "CONFIG\_SYS\_TEXT\_BASE = 0x00000000" > $(obj)board/freescale/m54451evb/config.tmp ; \          cp $(obj)board/freescale/m54451evb/u-boot.spa $(obj)board/freescale/m54451evb/u-boot.lds ; \      fi; \      if [ "$${FLASH}" = "STMICRO" ] ; then \          echo "#define CONFIG\_CF\_SBF"    >> $(obj)include/config.h ; \          echo "#define CONFIG\_SYS\_STMICRO\_BOOT"  >> $(obj)include/config.h ; \          echo "CONFIG\_SYS\_TEXT\_BASE = 0x47E00000" > $(obj)board/freescale/m54451evb/config.tmp ; \          cp $(obj)board/freescale/m54451evb/u-boot.stm $(obj)board/freescale/m54451evb/u-boot.lds ; \      fi; \      echo "#define CONFIG\_SYS\_INPUT\_CLKSRC 24000000" >> $(obj)include/config.h ;      @$(MKCONFIG) -n $@ -a M54451EVB m68k mcf5445x m54451evb freescale    M54455EVB\_config \  M54455EVB\_atmel\_config \  M54455EVB\_intel\_config \  M54455EVB\_a33\_config \  M54455EVB\_a66\_config \  M54455EVB\_i33\_config \  M54455EVB\_i66\_config \  M54455EVB\_stm33\_config :    unconfig      @case "$@" in \      M54455EVB\_config)       FLASH=ATMEL; FREQ=33333333;; \      M54455EVB\_atmel\_config)     FLASH=ATMEL; FREQ=33333333;; \      M54455EVB\_intel\_config)     FLASH=INTEL; FREQ=33333333;; \      M54455EVB\_a33\_config)       FLASH=ATMEL; FREQ=33333333;; \      M54455EVB\_a66\_config)       FLASH=ATMEL; FREQ=66666666;; \      M54455EVB\_i33\_config)       FLASH=INTEL; FREQ=33333333;; \      M54455EVB\_i66\_config)       FLASH=INTEL; FREQ=66666666;; \      M54455EVB\_stm33\_config)     FLASH=STMICRO; FREQ=33333333;; \      esac; \      if [ "$${FLASH}" = "INTEL" ] ; then \          echo "#define CONFIG\_SYS\_INTEL\_BOOT" >> $(obj)include/config.h ; \          echo "CONFIG\_SYS\_TEXT\_BASE = 0x00000000" > $(obj)board/freescale/m54455evb/config.tmp ; \          cp $(obj)board/freescale/m54455evb/u-boot.int $(obj)board/freescale/m54455evb/u-boot.lds ; \      fi; \      if [ "$${FLASH}" = "ATMEL" ] ; then \          echo "#define CONFIG\_SYS\_ATMEL\_BOOT"    >> $(obj)include/config.h ; \          echo "CONFIG\_SYS\_TEXT\_BASE = 0x04000000" > $(obj)board/freescale/m54455evb/config.tmp ; \          cp $(obj)board/freescale/m54455evb/u-boot.atm $(obj)board/freescale/m54455evb/u-boot.lds ; \      fi; \      if [ "$${FLASH}" = "STMICRO" ] ; then \          echo "#define CONFIG\_CF\_SBF"    >> $(obj)include/config.h ; \          echo "#define CONFIG\_SYS\_STMICRO\_BOOT"  >> $(obj)include/config.h ; \          echo "CONFIG\_SYS\_TEXT\_BASE = 0x4FE00000" > $(obj)board/freescale/m54455evb/config.tmp ; \          cp $(obj)board/freescale/m54455evb/u-boot.stm $(obj)board/freescale/m54455evb/u-boot.lds ; \      fi; \      echo "#define CONFIG\_SYS\_INPUT\_CLKSRC $${FREQ}" >> $(obj)include/config.h ; \      $(XECHO) "... with $${FREQ}Hz input clock"      @$(MKCONFIG) -n $@ -a M54455EVB m68k mcf5445x m54455evb freescale    M5475AFE\_config \  M5475BFE\_config \  M5475CFE\_config \  M5475DFE\_config \  M5475EFE\_config \  M5475FFE\_config \  M5475GFE\_config :   unconfig      @case "$@" in \      M5475AFE\_config)    BOOT=2;CODE=0;VID=0;USB=0;RAM=64;RAM1=0;; \      M5475BFE\_config)    BOOT=2;CODE=16;VID=0;USB=0;RAM=64;RAM1=0;; \      M5475CFE\_config)    BOOT=2;CODE=16;VID=1;USB=1;RAM=64;RAM1=0;; \      M5475DFE\_config)    BOOT=2;CODE=0;VID=0;USB=1;RAM=64;RAM1=0;; \      M5475EFE\_config)    BOOT=2;CODE=0;VID=1;USB=1;RAM=64;RAM1=0;; \      M5475FFE\_config)    BOOT=2;CODE=32;VID=1;USB=1;RAM=64;RAM1=64;; \      M5475GFE\_config)    BOOT=4;CODE=0;VID=0;USB=0;RAM=64;RAM1=0;; \      esac; \      echo "#define CONFIG\_SYS\_BUSCLK 133333333" > $(obj)include/config.h ; \      echo "#define CONFIG\_SYS\_BOOTSZ $${BOOT}" >> $(obj)include/config.h ; \      echo "#define CONFIG\_SYS\_DRAMSZ $${RAM}" >> $(obj)include/config.h ; \      if [ "$${RAM1}" != "0" ] ; then \          echo "#define CONFIG\_SYS\_DRAMSZ1    $${RAM1}" >> $(obj)include/config.h ; \      fi; \      if [ "$${CODE}" != "0" ] ; then \          echo "#define CONFIG\_SYS\_NOR1SZ $${CODE}" >> $(obj)include/config.h ; \      fi; \      if [ "$${VID}" == "1" ] ; then \          echo "#define CONFIG\_SYS\_VIDEO" >> $(obj)include/config.h ; \      fi; \      if [ "$${USB}" == "1" ] ; then \          echo "#define CONFIG\_SYS\_USBCTRL" >> $(obj)include/config.h ; \      fi      @$(MKCONFIG) -n $@ -a M5475EVB m68k mcf547x\_8x m547xevb freescale    M5485AFE\_config \  M5485BFE\_config \  M5485CFE\_config \  M5485DFE\_config \  M5485EFE\_config \  M5485FFE\_config \  M5485GFE\_config \  M5485HFE\_config :   unconfig      @case "$@" in \      M5485AFE\_config)    BOOT=2;CODE=0;VID=0;USB=0;RAM=64;RAM1=0;; \      M5485BFE\_config)    BOOT=2;CODE=16;VID=0;USB=0;RAM=64;RAM1=0;; \      M5485CFE\_config)    BOOT=2;CODE=16;VID=1;USB=1;RAM=64;RAM1=0;; \      M5485DFE\_config)    BOOT=2;CODE=0;VID=0;USB=1;RAM=64;RAM1=0;; \      M5485EFE\_config)    BOOT=2;CODE=0;VID=1;USB=1;RAM=64;RAM1=0;; \      M5485FFE\_config)    BOOT=2;CODE=32;VID=1;USB=1;RAM=64;RAM1=64;; \      M5485GFE\_config)    BOOT=4;CODE=0;VID=0;USB=0;RAM=64;RAM1=0;; \      M5485HFE\_config)    BOOT=2;CODE=16;VID=1;USB=0;RAM=64;RAM1=0;; \      esac; \      echo "#define CONFIG\_SYS\_BUSCLK 100000000" > $(obj)include/config.h ; \      echo "#define CONFIG\_SYS\_BOOTSZ $${BOOT}" >> $(obj)include/config.h ; \      echo "#define CONFIG\_SYS\_DRAMSZ $${RAM}" >> $(obj)include/config.h ; \      if [ "$${RAM1}" != "0" ] ; then \          echo "#define CONFIG\_SYS\_DRAMSZ1    $${RAM1}" >> $(obj)include/config.h ; \      fi; \      if [ "$${CODE}" != "0" ] ; then \          echo "#define CONFIG\_SYS\_NOR1SZ $${CODE}" >> $(obj)include/config.h ; \      fi; \      if [ "$${VID}" == "1" ] ; then \          echo "#define CONFIG\_SYS\_VIDEO" >> $(obj)include/config.h ; \      fi; \      if [ "$${USB}" == "1" ] ; then \          echo "#define CONFIG\_SYS\_USBCTRL" >> $(obj)include/config.h ; \      fi      @$(MKCONFIG) -n $@ -a M5485EVB m68k mcf547x\_8x m548xevb freescale    #========================================================================  # ARM  #========================================================================    xtract\_omap1610xxx = $(subst \_cs0boot,,$(subst \_cs3boot,,$(subst \_cs\_autoboot,,$(subst \_config,,$1))))    omap1610inn\_config \  omap1610inn\_cs0boot\_config \  omap1610inn\_cs3boot\_config \  omap1610inn\_cs\_autoboot\_config \  omap1610h2\_config \  omap1610h2\_cs0boot\_config \  omap1610h2\_cs3boot\_config \  omap1610h2\_cs\_autoboot\_config:  unconfig      @mkdir -p $(obj)include      @if [ "$(findstring \_cs0boot\_, $@)" ] ; then \          echo "#define CONFIG\_CS0\_BOOT" >> .$(obj)include/config.h ; \      elif [ "$(findstring \_cs\_autoboot\_, $@)" ] ; then \          echo "#define CONFIG\_CS\_AUTOBOOT" >> $(obj)include/config.h ; \      else \          echo "#define CONFIG\_CS3\_BOOT" >> $(obj)include/config.h ; \      fi;      @$(MKCONFIG) -n $@ -a $(call xtract\_omap1610xxx,$@) arm arm926ejs omap1610inn ti omap    omap730p2\_config \  omap730p2\_cs0boot\_config \  omap730p2\_cs3boot\_config :  unconfig      @mkdir -p $(obj)include      @if [ "$(findstring \_cs0boot\_, $@)" ] ; then \          echo "#define CONFIG\_CS0\_BOOT" >> $(obj)include/config.h ; \      else \          echo "#define CONFIG\_CS3\_BOOT" >> $(obj)include/config.h ; \      fi;      @$(MKCONFIG) -n $@ -a omap730p2 arm arm926ejs omap730p2 ti omap    spear300\_config \  spear310\_config \  spear320\_config :   unconfig      @$(MKCONFIG) -n $@ -t $@ spear3xx arm arm926ejs $(@:\_config=) spear spear    spear600\_config :   unconfig      @$(MKCONFIG) -n $@ -t $@ spear6xx arm arm926ejs $(@:\_config=) spear spear    SX1\_stdout\_serial\_config \  SX1\_config:     unconfig      @mkdir -p $(obj)include      @if [ "$(findstring \_stdout\_serial\_, $@)" ] ; then \          echo "#undef CONFIG\_STDOUT\_USBTTY" >> $(obj)include/config.h ; \      else \          echo "#define CONFIG\_STDOUT\_USBTTY" >> $(obj)include/config.h ; \      fi;      @$(MKCONFIG) -n $@ SX1 arm arm925t sx1    tx25\_config : unconfig      @echo "CONFIG\_NAND\_U\_BOOT = y" >> $(obj)include/config.mk      @$(MKCONFIG) $@ arm arm926ejs tx25 karo mx25    ti8168\_evm\_config   \  ti8168\_evm\_config\_nand  \  ti8168\_evm\_config\_nor   \  ti8168\_evm\_config\_spi   \  ti8168\_evm\_min\_ocmc \  ti8168\_evm\_min\_sd:  unconfig      @mkdir -p $(obj)include      @echo "#define CONFIG\_TI81XX"   >>$(obj)include/config.h      @echo "#define CONFIG\_TI816X"   >>$(obj)include/config.h      @if [ "$(findstring \_nand,$@)" ] ; then \          echo "#define CONFIG\_SYS\_NO\_FLASH"    >>$(obj)include/config.h ; \          echo "#define CONFIG\_NAND\_ENV"    >>$(obj)include/config.h ; \          echo "Setting up TI8168 NAND build with ENV in NAND..." ; \      elif [ "$(findstring \_nor,$@)" ] ; then \          echo "#define CONFIG\_NOR"    >>$(obj)include/config.h ; \          echo "#define CONFIG\_NOR\_BOOT"  >>$(obj)include/config.h ; \          echo "Setting up TI8168 NOR build with ENV in NOR..." ; \      elif [ "$(findstring \_spi,$@)" ] ; then \          echo "#define CONFIG\_SYS\_NO\_FLASH"    >>$(obj)include/config.h ; \          echo "#define CONFIG\_SPI\_ENV"    >>$(obj)include/config.h ; \          echo "#define CONFIG\_TI81XX\_SPI\_BOOT"   >>$(obj)include/config.h ; \          echo "Setting up TI8168 SPI build with ENV in SPI..." ; \      elif [ "$(findstring \_sd,$@)" ] ; then \          echo "#define CONFIG\_SYS\_NO\_FLASH"    >>$(obj)include/config.h ; \          echo "#define CONFIG\_SD\_BOOT"    >>$(obj)include/config.h ; \          echo "TI\_IMAGE = u-boot.min.sd" >>$(obj)board/ti/ti8168/config.tmp; \          echo "Setting up TI8168 SD boot minimal build..." ; \      elif [ "$(findstring \_ocmc,$@)" ] ; then \          echo "#define CONFIG\_SYS\_NO\_FLASH"    >>$(obj)include/config.h ; \          echo "#define CONFIG\_MINIMAL"    >>$(obj)include/config.h ; \          echo "CONFIG\_SYS\_TEXT\_BASE = 0x40410000" >>$(obj)board/ti/ti8168/config.tmp; \          echo "Setting up TI8168 minimal build..." ; \      else    \          echo "#define CONFIG\_SYS\_NO\_FLASH"    >>$(obj)include/config.h ; \          echo "#define CONFIG\_NAND\_ENV"    >>$(obj)include/config.h ; \          echo "Setting up TI8168 default build with NAND..." ; \      fi;      @$(MKCONFIG) -a ti8168\_evm arm armv7 ti8168 ti ti81xx    #########################################################################  ## XScale Systems  #########################################################################    pdnb3\_config \  scpu\_config:    unconfig      @mkdir -p $(obj)include      @if [ "$(findstring scpu\_,$@)" ] ; then \          echo "#define CONFIG\_SCPU"  >>$(obj)include/config.h ; \      fi      @$(MKCONFIG) -n $@ -a pdnb3 arm ixp pdnb3 prodrive    #########################################################################  ## ARM1136 Systems  #########################################################################    apollon\_config      : unconfig      @mkdir -p $(obj)include      @echo "#define CONFIG\_ONENAND\_U\_BOOT" > $(obj)include/config.h      @echo "CONFIG\_ONENAND\_U\_BOOT = y" >> $(obj)include/config.mk      @$(MKCONFIG) $@ arm arm1136 apollon - omap24xx    imx31\_phycore\_eet\_config \  imx31\_phycore\_config    : unconfig      @mkdir -p $(obj)include      @if [ -n "$(findstring \_eet\_,$@)" ]; then           \          echo "#define CONFIG\_IMX31\_PHYCORE\_EET" >> $(obj)include/config.h;    \      fi      @$(MKCONFIG) -n $@ -a imx31\_phycore arm arm1136 imx31\_phycore - mx31    mx31pdk\_config \  mx31pdk\_nand\_config : unconfig      @mkdir -p $(obj)include      @if [ -n "$(findstring \_nand\_,$@)" ]; then                  \          echo "#define CONFIG\_NAND\_U\_BOOT" >> $(obj)include/config.h;      \      else                                        \          echo "#define CONFIG\_SKIP\_LOWLEVEL\_INIT" >> $(obj)include/config.h;   \      fi      @$(MKCONFIG) -n $@ -a mx31pdk arm arm1136 mx31pdk freescale mx31    #########################################################################  ## ARM1176 Systems  #########################################################################  smdk6400\_noUSB\_config   \  smdk6400\_config :   unconfig      @mkdir -p $(obj)include $(obj)board/samsung/smdk6400      @mkdir -p $(obj)nand\_spl/board/samsung/smdk6400      @echo "#define CONFIG\_NAND\_U\_BOOT" > $(obj)include/config.h      @echo "CONFIG\_NAND\_U\_BOOT = y" >> $(obj)include/config.mk      @if [ -z "$(findstring smdk6400\_noUSB\_config,$@)" ]; then           \          echo "RAM\_TEXT = 0x57e00000" >> $(obj)board/samsung/smdk6400/config.tmp;\      else                                        \          echo "RAM\_TEXT = 0xc7e00000" >> $(obj)board/samsung/smdk6400/config.tmp;\      fi      @$(MKCONFIG) smdk6400 arm arm1176 smdk6400 samsung s3c64xx      @echo "CONFIG\_NAND\_U\_BOOT = y" >> $(obj)include/config.mk    #########################################################################  #########################################################################    clean:      @rm -f $(obj)examples/standalone/82559\_eeprom             \             $(obj)examples/standalone/atmel\_df\_pow2            \             $(obj)examples/standalone/eepro100\_eeprom          \             $(obj)examples/standalone/hello\_world              \             $(obj)examples/standalone/interrupt            \             $(obj)examples/standalone/mem\_to\_mem\_idma2intr         \             $(obj)examples/standalone/sched                \             $(obj)examples/standalone/smc911{11,x}\_eeprom          \             $(obj)examples/standalone/test\_burst           \             $(obj)examples/standalone/timer      @rm -f $(obj)examples/api/demo{,.bin}      @rm -f $(obj)tools/bmp\_logo    $(obj)tools/easylogo/easylogo  \             $(obj)tools/env/{fw\_printenv,fw\_setenv}            \             $(obj)tools/envcrc                     \             $(obj)tools/gdb/{astest,gdbcont,gdbsend}           \             $(obj)tools/gen\_eth\_addr    $(obj)tools/img2srec       \             $(obj)tools/mkimage     $(obj)tools/mpc86x\_clk     \             $(obj)tools/ncb         $(obj)tools/ubsha1      @rm -f $(obj)board/cray/L1/{bootscript.c,bootscript.image}    \             $(obj)board/matrix\_vision/\*/bootscript.img         \             $(obj)board/voiceblue/eeprom                   \             $(obj)u-boot.lds                       \             $(obj)arch/blackfin/cpu/bootrom-asm-offsets.[chs]      \             $(obj)arch/blackfin/cpu/init.{lds,elf}      @rm -f $(obj)include/bmp\_logo.h      @rm -f $(obj)lib/asm-offsets.s      @rm -f $(obj)include/generated/asm-offsets.h      @rm -f $(obj)$(CPUDIR)/$(SOC)/asm-offsets.s      @rm -f $(obj)nand\_spl/{u-boot.lds,u-boot-nand\_spl.lds,u-boot-spl,u-boot-spl.map,System.map}      @rm -f $(obj)onenand\_ipl/onenand-{ipl,ipl.bin,ipl.map}      @rm -f $(obj)mmc\_spl/{u-boot.lds,u-boot-spl,u-boot-spl.map,u-boot-spl.bin,u-boot-mmc-spl.bin}      @rm -f $(ONENAND\_BIN)      @rm -f $(obj)onenand\_ipl/u-boot.lds      @rm -f $(obj)spl/{u-boot-spl,u-boot-spl.bin,u-boot-spl.lds,u-boot-spl.map}      @rm -f $(TIMESTAMP\_FILE) $(VERSION\_FILE)      @find $(OBJTREE) -type f \          \( -name 'core' -o -name '\*.bak' -o -name '\*~' \          -o -name '\*.o'  -o -name '\*.a' -o -name '\*.exe' \) -print \          | xargs rm -f    clobber:    clean      @find $(OBJTREE) -type f \( -name '\*.depend' \          -o -name '\*.srec' -o -name '\*.bin' -o -name u-boot.img \) \          -print0 \          | xargs -0 rm -f      @rm -f $(OBJS) $(obj)\*.bak $(obj)ctags $(obj)etags $(obj)TAGS \          $(obj)cscope.\* $(obj)\*.\*~      @rm -f $(obj)u-boot $(obj)u-boot.map $(obj)u-boot.hex $(ALL-y)      @rm -f $(obj)u-boot.kwb      @rm -f $(obj)u-boot.imx      @rm -f $(obj)u-boot.ubl      @rm -f $(obj)tools/{env/crc32.c,inca-swap-bytes}      @rm -f $(obj)arch/powerpc/cpu/mpc824x/bedbug\_603e.c      @rm -fr $(obj)include/asm/proc $(obj)include/asm/arch $(obj)include/asm      @rm -fr $(obj)include/generated      @[ ! -d $(obj)nand\_spl ] || find $(obj)nand\_spl -name "\*" -type l -print | xargs rm -f      @[ ! -d $(obj)onenand\_ipl ] || find $(obj)onenand\_ipl -name "\*" -type l -print | xargs rm -f      @[ ! -d $(obj)mmc\_spl ] || find $(obj)mmc\_spl -name "\*" -type l -print | xargs rm -f    mrproper \  distclean:  clobber unconfig  ifneq ($(OBJTREE),$(SRCTREE))      rm -rf $(obj)\*  endif    backup:      F=`basename $(TOPDIR)` ; cd .. ; \      gtar --force-local -zcvf `LC\_ALL=C date "+$$F-%Y-%m-%d-%T.tar.gz"` $$F    ######################################################################### |

**config.mk 详解：**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191  192  193  194  195  196  197  198  199  200  201  202  203  204  205  206  207  208  209  210  211  212  213  214  215  216  217  218  219  220  221  222  223  224  225  226  227  228  229  230  231  232  233  234  235  236  237  238  239  240  241  242  243  244  245  246  247  248  249  250  251  252  253  254  255  256  257  258  259  260  261  262  263  264  265  266  267  268  269  270  271  272  273  274  275  276  277  278  279  280  281  282  283  284  285  286  287  288  289  290  291  292  293  294  295  296  297  298  299  300  301  302  303  304  305  306  307  308  309  310  311  312  313  314  315  316  317  318  319  320  321  322  323  324  325  326  327  328  329  330  331  332  333  334  335  336  337  338  339  340  341  342  343  344  345  346  347  348  349  350  351  352  353  354  355  356  357  358  359  360  361  362  363  364  365  366  367  368  369  370  371  372  373  374  375  376  377  378  379  380  381  382  383  384  385  386  387  388  389  390  391  392  393  394  395  396  397  398  399 | #  # (C) Copyright 2000-2006  # Wolfgang Denk, DENX Software Engineering, wd@denx.de.  #  # See file CREDITS for list of people who contributed to this  # project.  #  # This program is free software; you can redistribute it and/or  # modify it under the terms of the GNU General Public License as  # published by the Free Software Foundation; either version 2 of  # the License, or (at your option) any later version.  #  # This program is distributed in the hope that it will be useful,  # but WITHOUT ANY WARRANTY; without even the implied warranty of  # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the  # GNU General Public License for more details.  #  # You should have received a copy of the GNU General Public License  # along with this program; if not, write to the Free Software  # Foundation, Inc., 59 Temple Place, Suite 330, Boston,  # MA 02111-1307 USA  #    #########################################################################  #当前     CURDIR  = ./  #       SRCTREE = ./  #则 dir 为空  #########################################################################  ifeq ($(CURDIR),$(SRCTREE))  dir :=  else  dir := $(subst $(SRCTREE)/,,$(CURDIR))  endif    #########################################################################  #当前     OBJTREE = ./am335x  #       SRCTREE = ./  #判断变量OBJTREE,SRCTREE是否相等,其不相同，则条件为真  #查找CONFIG\_SPL\_BUILD是否定义为y，在autoconf.mk(./am335x/include/autoconf.mk)中,并没有这个定义  #则  obj := $(if $(dir),$(OBJTREE)/$(dir)/,$(OBJTREE)/) ->    obj = ./am335x/  #   src = ./  # dir 为空  # SPLTREE 为 ./am335x/spl  # 然后创建 ./am335x 目录  ## -p 表示如果目录已存在，并不会产生错误  #########################################################################  ifneq ($(OBJTREE),$(SRCTREE))  # Create object files for SPL in a separate directory  ifeq ($(CONFIG\_SPL\_BUILD),y)  obj := $(if $(dir),$(SPLTREE)/$(dir)/,$(SPLTREE)/)  else  obj := $(if $(dir),$(OBJTREE)/$(dir)/,$(OBJTREE)/)  endif  src := $(if $(dir),$(SRCTREE)/$(dir)/,$(SRCTREE)/)    $(shell mkdir -p $(obj))  else  # Create object files for SPL in a separate directory  ifeq ($(CONFIG\_SPL\_BUILD),y)  obj := $(if $(dir),$(SPLTREE)/$(dir)/,$(SPLTREE)/)    $(shell mkdir -p $(obj))  else  obj :=  endif  src :=  endif    # clean the slate ...  PLATFORM\_RELFLAGS =  PLATFORM\_CPPFLAGS =  PLATFORM\_LDFLAGS =    #########################################################################    HOSTCFLAGS  = -Wall -Wstrict-prototypes -O2 -fomit-frame-pointer \            $(HOSTCPPFLAGS)  HOSTSTRIP   = strip    #  # Mac OS X / Darwin's C preprocessor is Apple specific.  It  # generates numerous errors and warnings.  We want to bypass it  # and use GNU C's cpp.  To do this we pass the -traditional-cpp  # option to the compiler.  Note that the -traditional-cpp flag  # DOES NOT have the same semantics as GNU C's flag, all it does  # is invoke the GNU preprocessor in stock ANSI/ISO C fashion.  #  # Apple's linker is similar, thanks to the new 2 stage linking  # multiple symbol definitions are treated as errors, hence the  # -multiply\_defined suppress option to turn off this error.  #  #当前 HOSTOS 为linux,条件为假，不执行条件中的代码  #得 HOSTCC       = gcc  ifeq ($(HOSTOS),darwin)  # get major and minor product version (e.g. '10' and '6' for Snow Leopard)  DARWIN\_MAJOR\_VERSION    = $(shell sw\_vers -productVersion | cut -f 1 -d '.')  DARWIN\_MINOR\_VERSION    = $(shell sw\_vers -productVersion | cut -f 2 -d '.')    os\_x\_before = $(shell if [ $(DARWIN\_MAJOR\_VERSION) -le $(1) -a \      $(DARWIN\_MINOR\_VERSION) -le $(2) ] ; then echo "$(3)"; else echo "$(4)"; fi ;)    # Snow Leopards build environment has no longer restrictions as described above  HOSTCC       = $(call os\_x\_before, 10, 5, "cc", "gcc")  HOSTCFLAGS  += $(call os\_x\_before, 10, 4, "-traditional-cpp")  HOSTLDFLAGS += $(call os\_x\_before, 10, 5, "-multiply\_defined suppress")  else  HOSTCC      = gcc  endif    #当前 HOSTOS 为linux,条件为假，不执行条件中的代码  ifeq ($(HOSTOS),cygwin)  HOSTCFLAGS  += -ansi  endif    # We build some files with extra pedantic flags to try to minimize things  # that won't build on some weird host compiler -- though there are lots of  # exceptions for files that aren't complaint.  #########################################################################  #HOSTCFLAGS\_NOPED是利用filter-out函数从HOSTCFLAGS中过滤掉-pedantic选项  #而HOSTCFLAGS追加上-pedantic选项  #########################################################################  HOSTCFLAGS\_NOPED = $(filter-out -pedantic,$(HOSTCFLAGS))  HOSTCFLAGS  += -pedantic    #########################################################################  #  # Option checker (courtesy linux kernel) to ensure  # only supported compiler options are used  #  #########################################################################  #CC = $(CROSS\_COMPILE)gcc  #CROSS\_COMPILE = arm-arago-linux-gnueabi-  #CC = arm-arago-linux-gnueabi-gcc  # 函数 cc-option = $(CFLAGS) $(1) -S -o /dev/null -xc /dev/null > /dev/null 2>&1; then echo "$(1)";  #########################################################################  cc-option = $(shell if $(CC) $(CFLAGS) $(1) -S -o /dev/null -xc /dev/null \          > /dev/null 2>&1; then echo "$(1)"; else echo "$(2)"; fi ;)    #  # Include the make variables (CC, etc...)  #  AS  = $(CROSS\_COMPILE)as  LD  = $(CROSS\_COMPILE)ld  CC  = $(CROSS\_COMPILE)gcc  CPP = $(CC) -E  AR  = $(CROSS\_COMPILE)ar  NM  = $(CROSS\_COMPILE)nm  LDR = $(CROSS\_COMPILE)ldr  STRIP   = $(CROSS\_COMPILE)strip  OBJCOPY = $(CROSS\_COMPILE)objcopy  OBJDUMP = $(CROSS\_COMPILE)objdump  RANLIB  = $(CROSS\_COMPILE)RANLIB    #########################################################################  #包含文件   ./am335x/include/autoconf.mk,编译时需要用到的一些宏定义;  #           ./am335x/include/config.mk,开发板的相关信息  sinclude $(OBJTREE)/include/autoconf.mk  sinclude $(OBJTREE)/include/config.mk    # Some architecture config.mk files need to know what CPUDIR is set to,  # so calculate CPUDIR before including ARCH/SOC/CPU config.mk files.  # Check if arch/$ARCH/cpu/$CPU exists, otherwise assume arch/$ARCH/cpu contains  # CPU-specific code.  #########################################################################/\*  #   ARCH    = arm  #   CPU     = armv7  # 得CPUDIR   = arch/arm/cpu/armv7  #########################################################################\*/  CPUDIR=arch/$(ARCH)/cpu/$(CPU)    #########################################################################/\*  #   SRCTREE = ./( uboot的根目录)  #   $(SRCTREE)/$(CPUDIR) = ./arch/arm/cpu/armv7  #存在该目录，条件为假，条件中的代码不会被执行  #########################################################################\*/  ifneq ($(SRCTREE)/$(CPUDIR),$(wildcard $(SRCTREE)/$(CPUDIR)))  CPUDIR=arch/$(ARCH)/cpu  endif    #########################################################################/\*  #   TOPDIR  = ./( uboot的根目录)  #   ARCH    = arm  #   CPUDIR  = arch/arm/cpu/armv7  #包含文件 $(TOPDIR)/arch/$(ARCH)/config.mk  -> ./arch/arm/config.mk  #包含文件 $(TOPDIR)/$(CPUDIR)/config.mk         -> ./arch/arm/cpu/armv7/config.mk  #########################################################################\*/  sinclude $(TOPDIR)/arch/$(ARCH)/config.mk   # include architecture dependend rules  sinclude $(TOPDIR)/$(CPUDIR)/config.mk      # include  CPU  specific rules    #########################################################################/\*  #判断是否定义了 SOC，当前 SOC 为 ti81xx，条件为 真，  #则包含文件 $(TOPDIR)/$(CPUDIR)/$(SOC)/config.mk -> ./arch/arm/cpu/armv7/ti81xx/config.mk  #   TOPDIR  = ./( uboot的根目录)  #   SOC     = ti81xx  #   CPUDIR  = arch/arm/cpu/armv7  #########################################################################\*/  ifdef   SOC  sinclude $(TOPDIR)/$(CPUDIR)/$(SOC)/config.mk   # include  SoC  specific rules  endif    #########################################################################/\*  #判断是否定义了 VENDOR，当前 VENDOR 为 ti，条件为 真，  #则得 BOARDDIR = ti/am335x  #   VENDOR  = ti  #   BOARD   = am335x  #########################################################################\*/  ifdef   VENDOR  BOARDDIR = $(VENDOR)/$(BOARD)  else  BOARDDIR = $(BOARD)  endif    #########################################################################/\*  #判断是否定义了 BOARD，当前 BOARD 为 am335x，条件为 真，  #则得 BOARDDIR = ti/am335x  #   BOARD   = am335x  #   TOPDIR  = ./( uboot的根目录)  #包含文件 $(TOPDIR)/board/$(BOARDDIR)/config.mk     -> ./board/ti/am335x/config.mk  # !!!但是找不到相应的文件!!!  #########################################################################\*/  ifdef   BOARD  sinclude $(TOPDIR)/board/$(BOARDDIR)/config.mk  # include board specific rules  endif    #########################################################################  #########################################################################/\*  #   MAKEFLAGS       = wp -- $(MAKEOVERRIDES)  #   MAKEOVERRIDES   = ${-\*-command-variables-\*-}  #   -\*-command-variables-\*- := O=am335x  #判断字符串 s 是否为空, s = O=am335x,s不为空,条件为假，则 ARFLAGS = crv  #########################################################################\*/  ifneq (,$(findstring s,$(MAKEFLAGS)))  ARFLAGS = cr  else  ARFLAGS = crv  endif    #PLATFORM\_RELFLAGS =  -fno-common -ffixed-r8 -msoft-float $(call cc-option,-mshort-load-bytes, $(call cc-option,-malignment-traps,))  RELFLAGS= $(PLATFORM\_RELFLAGS)  DBGFLAGS= -g # -DDEBUG  OPTFLAGS= -Os #-fomit-frame-pointer    OBJCFLAGS += --gap-fill=0xff    #gccincdir = arm-arago-linux-gnueabi-gcc -print-file-name=include  gccincdir := $(shell $(CC) -print-file-name=include)    #CPPFLAGS = -g  -Os   -fno-common -ffixed-r8 -msoft-float   -D\_\_KERNEL\_\_  CPPFLAGS := $(DBGFLAGS) $(OPTFLAGS) $(RELFLAGS)     \      -D\_\_KERNEL\_\_    # Enable garbage collection of un-used sections for SPL  #########################################################################/\*  #CONFIG\_SPL\_BUILD = y  #判断 CONFIG\_SPL\_BUILD 是否为 y，而 CONFIG\_SPL\_BUILD=y，条件为真，执行条件下的代码  #########################################################################\*/  ifeq ($(CONFIG\_SPL\_BUILD),y)  CPPFLAGS += -ffunction-sections -fdata-sections  LDFLAGS\_FINAL += --gc-sections  endif    #########################################################################/\*  #CONFIG\_SYS\_TEXT\_BASE=0x80800000  #判断 CONFIG\_SYS\_TEXT\_BASE 是否为空，而 CONFIG\_SYS\_TEXT\_BASE=0x80800000，条件为真，执行条件下的代码  #########################################################################\*/  ifneq ($(CONFIG\_SYS\_TEXT\_BASE),)  CPPFLAGS += -DCONFIG\_SYS\_TEXT\_BASE=$(CONFIG\_SYS\_TEXT\_BASE)  endif    #########################################################################/\*  #CONFIG\_SPL\_TEXT\_BASE=0x402F0400  #判断 CONFIG\_SPL\_TEXT\_BASE 是否为空，而 CONFIG\_SPL\_TEXT\_BASE=0x402F0400，条件为真，执行条件下的代码  #########################################################################\*/  ifneq ($(CONFIG\_SPL\_TEXT\_BASE),)  CPPFLAGS += -DCONFIG\_SPL\_TEXT\_BASE=$(CONFIG\_SPL\_TEXT\_BASE)  endif    #########################################################################/\*  #CONFIG\_SPL\_BUILD = y  #判断 CONFIG\_SPL\_BUILD 是否为 y，而 CONFIG\_SPL\_BUILD=y，条件为真，执行条件下的代码  #########################################################################\*/  ifeq ($(CONFIG\_SPL\_BUILD),y)  CPPFLAGS += -DCONFIG\_SPL\_BUILD  endif    #########################################################################/\*  #判断 CONFIG\_SPL\_BUILD 是否为 空，而 RESET\_VECTOR\_ADDRESS 为空，条件为假，不执行条件下的代码  #########################################################################\*/  ifneq ($(RESET\_VECTOR\_ADDRESS),)  CPPFLAGS += -DRESET\_VECTOR\_ADDRESS=$(RESET\_VECTOR\_ADDRESS)  endif    #########################################################################  #当前     OBJTREE = ./am335x  #       SRCTREE = ./  #判断变量OBJTREE,SRCTREE是否相等,其不相同，则条件为真,执行条件下的代码  #########################################################################  ifneq ($(OBJTREE),$(SRCTREE))  CPPFLAGS += -I$(OBJTREE)/include2 -I$(OBJTREE)/include  endif    CPPFLAGS += -I$(TOPDIR)/include  CPPFLAGS += -fno-builtin -ffreestanding -nostdinc   \      -isystem $(gccincdir) -pipe $(PLATFORM\_CPPFLAGS)    #没有定义BUILD\_TAG，因此：CFLAGS := $(CPPFLAGS) -Wall -Wstrict-prototypes  ifdef BUILD\_TAG  CFLAGS := $(CPPFLAGS) -Wall -Wstrict-prototypes \      -DBUILD\_TAG='"$(BUILD\_TAG)"'  else  CFLAGS := $(CPPFLAGS) -Wall -Wstrict-prototypes  endif    CFLAGS += $(call cc-option,-fno-stack-protector)  # Some toolchains enable security related warning flags by default,  # but they don't make much sense in the u-boot world, so disable them.  CFLAGS += $(call cc-option,-Wno-format-nonliteral)  CFLAGS += $(call cc-option,-Wno-format-security)    # $(CPPFLAGS) sets -g, which causes gcc to pass a suitable -g<format>  # option to the assembler.  AFLAGS\_DEBUG :=    # turn jbsr into jsr for m68k  ifeq ($(ARCH),m68k)  ifeq ($(findstring 3.4,$(shell $(CC) --version)),3.4)  AFLAGS\_DEBUG := -Wa,-gstabs,-S  endif  endif    AFLAGS := $(AFLAGS\_DEBUG) -D\_\_ASSEMBLY\_\_ $(CPPFLAGS)    LDFLAGS += $(PLATFORM\_LDFLAGS)  LDFLAGS\_FINAL += -Bstatic    LDFLAGS\_u-boot += -T $(obj)u-boot.lds $(LDFLAGS\_FINAL)  ifneq ($(CONFIG\_SYS\_TEXT\_BASE),)  LDFLAGS\_u-boot += -Ttext $(CONFIG\_SYS\_TEXT\_BASE)  endif    LDFLAGS\_u-boot-spl += -T $(obj)u-boot-spl.lds $(LDFLAGS\_FINAL)  ifneq ($(CONFIG\_SPL\_TEXT\_BASE),)  LDFLAGS\_u-boot-spl += -Ttext $(CONFIG\_SPL\_TEXT\_BASE)  endif    # Location of a usable BFD library, where we define "usable" as  # "built for ${HOST}, supports ${TARGET}".  Sensible values are  # - When cross-compiling: the root of the cross-environment  # - Linux/ppc (native): /usr  # - NetBSD/ppc (native): you lose ... (must extract these from the  #   binutils build directory, plus the native and U-Boot include  #   files don't like each other)  #  # So far, this is used only by tools/gdb/Makefile.    ifeq ($(HOSTOS),darwin)  BFD\_ROOT\_DIR =      /usr/local/tools  else  ifeq ($(HOSTARCH),$(ARCH))  # native  BFD\_ROOT\_DIR =      /usr  else  #BFD\_ROOT\_DIR =     /LinuxPPC/CDK       # Linux/i386  #BFD\_ROOT\_DIR =     /usr/pkg/cross      # NetBSD/i386  BFD\_ROOT\_DIR =      /opt/powerpc  endif  endif    #########################################################################    export  HOSTCC HOSTCFLAGS HOSTLDFLAGS PEDCFLAGS HOSTSTRIP CROSS\_COMPILE \      AS LD CC CPP AR NM STRIP OBJCOPY OBJDUMP MAKE  export  CONFIG\_SYS\_TEXT\_BASE PLATFORM\_CPPFLAGS PLATFORM\_RELFLAGS CPPFLAGS CFLAGS AFLAGS    #########################################################################    # Allow boards to use custom optimize flags on a per dir/file basis  BCURDIR = $(subst $(SRCTREE)/,,$(CURDIR:$(obj)%=%))  ALL\_AFLAGS = $(AFLAGS) $(AFLAGS\_$(BCURDIR)/$(@F)) $(AFLAGS\_$(BCURDIR))  ALL\_CFLAGS = $(CFLAGS) $(CFLAGS\_$(BCURDIR)/$(@F)) $(CFLAGS\_$(BCURDIR))  $(obj)%.s:  %.S      $(CPP) $(ALL\_AFLAGS) -o $@ $<  $(obj)%.o:  %.S      $(CC)  $(ALL\_AFLAGS) -o $@ $< -c  $(obj)%.o:  %.c      $(CC)  $(ALL\_CFLAGS) -o $@ $< -c  $(obj)%.i:  %.c      $(CPP) $(ALL\_CFLAGS) -o $@ $< -c  $(obj)%.s:  %.c      $(CC)  $(ALL\_CFLAGS) -o $@ $< -c -S    #########################################################################    # If the list of objects to link is empty, just create an empty built-in.o  cmd\_link\_o\_target = $(if $(strip $1),\                $(LD) $(LDFLAGS) -r -o $@ $1,\                rm -f $@; $(AR) rcs $@ )    ######################################################################### |

**mkconfig 详解：**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191  192  193  194  195  196  197  198  199  200  201  202  203  204  205  206  207  208  209  210  211  212  213  214  215  216  217  218  219  220  221  222  223  224  225  226  227  228  229  230  231  232  233  234  235  236  237  238  239  240  241  242  243  244  245  246  247  248  249  250  251  252  253  254  255  256  257  258  259  260  261  262  263  264  265  266  267  268  269  270  271  272  273  274  275  276  277  278  279  280 | #!/bin/sh -e    # Script to create header files and links to configure  # U-Boot for a specific board.  #  # Parameters:  Target  Architecture  CPU  Board [VENDOR] [SOC]  #  # (C) 2002-2010 DENX Software Engineering, Wolfgang Denk <wd@denx.de>  #    APPEND=no   # Default: Create new config file  BOARD\_NAME=""   # Name to print in make output  TARGETS=""    arch=""  cpu=""  board=""  vendor=""  soc=""  options=""    #########################################################################  #我们执行脚本的命令是mkconfig -A am335x\_evm,$#表示的是参数的个数，$1表示的是第一个参数  #接下来就是使用egrep命令把单板名字$2的那行找出来，如果失败的话说明没有找到相应的单板，退出。  #如果找到的话，我们需要的信息都有了.  #set $(line) 这句话就是将line的信息设置成系统的位置变量  #line 就是在boards.cfg文件中am335x\_evm的那行，而-i表示忽略大小写  #在boards.cfg文件中，有  #Target          ARCH        CPU         Board name          Vendor          SoC         Options  #am335x\_evm      arm         armv7       am335x              ti             ti81xx  #  set ${line}  #  set也可用于在脚本内部给出其运行参数,所以这个时候参数就变为:  #"am335x\_evm      arm         armv7       am335x              ti             ti81xx"  #这个时候参数个数就变成6个了  #########################################################################  if [ \( $# -eq 2 \) -a \( "$1" = "-A" \) ] ; then      # Automatic mode      line=`egrep -i "^[[:space:]]\*${2}[[:space:]]" boards.cfg` || {          echo "make: \*\*\* No rule to make target \`$2\_config'.  Stop." >&2          exit 1      }        set ${line}      # add default board name if needed      [ $# = 3 ] && set ${line} ${1}  fi    #########################################################################  #环境变量$#表示传递给脚本的参数个数，这里的命令有6个参数，因此$#是6 。  #shift的作用是使$1=$2，$2=$3，$3=$4….，而原来的$1将丢失。因此while循环的作用是，  #依次处理传递给mkconfig脚本的选项。由于我们并没有传递给mkconfig任何的选项，  #因此while循环中的代码不起作用。  #########################################################################  #参数个数大于 0  while [ $# -gt 0 ] ; do      case "$1" in      --) shift ; break ;;      -a) shift ; APPEND=yes ;;      -n) shift ; BOARD\_NAME="${1%\_config}" ; shift ;;      -t) shift ; TARGETS="`echo $1 | sed 's:\_: :g'` ${TARGETS}" ; shift ;;      \*)  break ;;      esac  done    #参数个数少于 4 ,异常退出  [ $# -lt 4 ] && exit 1  #参数个数大于 7 ,异常退出  [ $# -gt 7 ] && exit 1    #########################################################################  #${1%\_config}的意思 $1 如果以 \_config 后缀结束的话，则要去掉 \_config 后缀。  #$1 (当前为 am335x\_evm），则 CONFIG\_NAME = am335x\_evm；  #假设 $1 为 am335x\_evm\_config，则 CONFIG\_NAME = am335x\_evm。  #########################################################################  # Strip all options and/or \_config suffixes  CONFIG\_NAME="${1%\_config}"    #如果  BOARD\_NAME 为空，则把 $1 参数赋值给 BOARD\_NAME  #（如果以 \_config 后缀结束的话，则要去掉 \_config 后缀）。  #当前 BOARD\_NAME = am335x\_evm  [ "${BOARD\_NAME}" ] || BOARD\_NAME="${1%\_config}"    #arch=arm  arch="$2"    #cpu=armv7  cpu="$3"    #board=am335x  if [ "$4" = "-" ] ; then      board=${BOARD\_NAME}  else      board="$4"  fi    #参数个数大于 4，且 参数5 并不为 “-”，则 vendor = ti  [ $# -gt 4 ] && [ "$5" != "-" ] && vendor="$5"    #参数个数大于 5，且 参数6 并不为 “-”，则 soc = ti81xx  [ $# -gt 5 ] && [ "$6" != "-" ] && soc="$6"    #当前参数只有 6 个，条件为假,这一段代码不会被执行  [ $# -gt 6 ] && [ "$7" != "-" ] && {      # check if we have a board config name in the options field      # the options field mave have a board config name and a list      # of options, both separated by a colon (':'); the options are      # separated by commas (',').      #      # Check for board name      tmp="${7%:\*}"      if [ "$tmp" ] ; then          CONFIG\_NAME="$tmp"      fi      # Check if we only have a colon...      if [ "${tmp}" != "$7" ] ; then          options=${7#\*:}          TARGETS="`echo ${options} | sed 's:,: :g'` ${TARGETS}"      fi  }    #########################################################################  #ARCH是在顶层makefile中定义的，在此刻还是为空的。  #如果ARCH已经有值了，那么就检测ARCH和arch是否匹配了.  #########################################################################  if [ "${ARCH}" -a "${ARCH}" != "${arch}" ]; then      echo "Failed: \$ARCH=${ARCH}, should be '${arch}' for ${BOARD\_NAME}" 1>&2      exit 1  fi    #########################################################################  #判断 options 是否为空，当前 options 为空，BOARD\_NAME 为 am335x\_evm,  #因此显示如下：Configuring for am335x\_evm board...  #########################################################################  if [ "$options" ] ; then      echo "Configuring for ${BOARD\_NAME} - Board: ${CONFIG\_NAME}, Options: ${options}"  else      echo "Configuring for ${BOARD\_NAME} board..."  fi    #########################################################################  #SRCTREE 为uboot根目录，设为 ： ./ （即当前目录，即uboot根目录）  #则 OBJTREE 为 SRCTREE目录下的am335x 目录，即为：./am335x/  #########################################################################  #  # Create link to architecture specific headers  #  if [ "$SRCTREE" != "$OBJTREE" ] ; then      #创建include目录：./am335x/include/      # -p 表示如果目录已存在，并不会产生错误      mkdir -p ${OBJTREE}/include        #创建include2目录：./am335x/include2/      mkdir -p ${OBJTREE}/include2        #进入到 ./am335x/include2/ 目录中      cd ${OBJTREE}/include2        #如果 ./am335x/include2 目录中有 asm 文件的话，则删除。      # -f 表示如果文件不存在，不提示      rm -f asm        #创建符号链接asm，其指向 ./arch/arm/include/asm      ln -s ${SRCTREE}/arch/${arch}/include/asm asm        #创建变量 LNPREFIX ，其赋值为 ：./arch/arm/include/asm      LNPREFIX=${SRCTREE}/arch/${arch}/include/asm/        #进入到 ./am335x/include/ 目录中      cd ../include        #创建 asm目录：./am335x/include/asm/      mkdir -p asm  else      cd ./include      rm -f asm      ln -s ../arch/${arch}/include/asm asm  fi    #当前路径为：./am335x/include/。  #删除 asm/arch 文件  rm -f asm/arch    #########################################################################  #判断 soc 是否为空，当前 soc = ti81xx，并不为空。  #因此创建符号链接 asm/arch，即 ./am335x/include/asm/arch 指向 ./arch/arm/include/asm/arch-ti81xx/ 目录  #########################################################################  if [ -z "${soc}" ] ; then      ln -s ${LNPREFIX}arch-${cpu} asm/arch  else      ln -s ${LNPREFIX}arch-${soc} asm/arch  fi    #########################################################################  #判断 变量 arch 是否为 arm，当前的arch为 arm，条件为真。  #当前路径为：./am335x/include/。  #删除 asm/proc 文件  #创建符号链接 asm/proc，即 ./am335x/include/asm/proc 指向 ./arch/arm/include/asm/proc-armv/ 目录.  #########################################################################  if [ "${arch}" = "arm" ] ; then      rm -f asm/proc      ln -s ${LNPREFIX}proc-armv asm/proc  fi    #########################################################################  #当前路径为：./am335x/include/。  #   > 表示如果文件不存在，则创建文件；如果文件存在，则清空文件  #   >>表示添加到文件的末尾  #########################################################################  #  # Create include file for Make  #  echo "ARCH   = ${arch}"  >  config.mk  echo "CPU    = ${cpu}"   >> config.mk  echo "BOARD  = ${board}" >> config.mk    #如果变量 vendor 不为空，则输入到文件末尾  [ "${vendor}" ] && echo "VENDOR = ${vendor}" >> config.mk    #如果变量 soc 不为空，则输入到文件末尾  [ "${soc}"    ] && echo "SOC    = ${soc}"    >> config.mk  #########################################################################  #最后得到文件 ./am335x/include/config.mk 内容如下：  #   ARCH   = arm  #   CPU    = armv7  #   BOARD  = am335x  #   VENDOR = ti  #   SOC    = ti81xx  #########################################################################    #判断 变量vendor 是否为空，当前变量vendor 为 ti，则条件为假，其中 board 为 am335x  #则 BOARDDIR = ti/am335x  # Assign board directory to BOARDIR variable  if [ -z "${vendor}" ] ; then      BOARDDIR=${board}  else      BOARDDIR=${vendor}/${board}  fi    #########################################################################  #判断 变量APPEND 是否为 yes，当前 APPEND为no，条件为假。  #则创建config.h 文件，如果该文件已存在，则清空。  #########################################################################  #  # Create board specific header file  #  if [ "$APPEND" = "yes" ]    # Append to existing config file  then      echo >> config.h  else      > config.h       # Create new config file  fi  echo "/\* Automatically generated - do not edit \*/" >>config.h    #当前 变量TARGETS 为空，其下面代码不会被执行  for i in ${TARGETS} ; do      i="`echo ${i} | sed '/=/ {s/=/  /;q; } ; { s/$/ 1/; }'`"      echo "#define CONFIG\_${i}" >>config.h ;  done    #将输入的内容追加到config.h中，直到出现“EOF”这样的标识为止。  cat << EOF >> config.h  #define CONFIG\_BOARDDIR board/$BOARDDIR  #include <config\_cmd\_defaults.h>  #include <config\_defaults.h>  #include <configs/${CONFIG\_NAME}.h>  #include <asm/config.h>  EOF  #########################################################################  #最后得到文件 ./am335x/include/config.h 内容如下：  #   /\* Automatically generated - do not edit \*/  #   #define CONFIG\_BOARDDIR board/ti/am335x  #   #include <config\_cmd\_defaults.h>  #   #include <config\_defaults.h>  #   #include <configs/am335x\_evm.h>  #   #include <asm/config.h>  #########################################################################      #正常退出  exit 0 |