

THUSC2023 Day2 题面

THUSC / THUWC 工程题补全计划

目录

组合逻辑电路部分	3
任务 1: 投票器	4
任务 2: 译码器	5
任务 3: 选择器	6
任务 4: 比较器	8
任务 5: 加法器	9
附加任务: 超前进位加法器	10
任务 6: ALU	11
时序逻辑电路部分	13
任务 1: 串行奇偶校验器	14
任务 2: 移位寄存器	15
任务 3: 可变模计数器	16
任务 4: 乘法器	17
任务 5: 寄存器堆	18
任务 6: 奋斗四小时, 手搓 CPU	20

组合逻辑电路部分

在这一部分中，你需要利用基础的门电路，完成一些简单的元件设计需求，并最终实现一个 ALU。

下面是一些约定：

- 在未说明的情况下，最大允许门延迟 T_{comb} 为 1000。关于最大门延迟的定义，请参照学习手册，下同。
- 在未说明的情况下，每个多输入逻辑门可以接受的最多输入信号个数 $MaxDegree$ 为 10。
- 在这部分中，你不应当，也不需要使⽤正边沿 D-触发器。可以使用的逻辑门仅限于 NOT, AND, OR, XOR, NAND, NOR, NXOR。
- 在输出中，不应当包含 x 状态。你可以利用一些操作消除 x 带来的影响，详见学习手册。
- 输入中涉及到的数字在输入信号中均以二进制从低到高的顺序给出。
- 功能表中的 x 和学习手册定义的特殊数字信号不同，代表的是 **取 0 或者 1 都成立** 的通配符。
- 更宏观的约定已经在学习手册中的 **答案格式** 中给出。

任务 1：投票器

最大允许门延迟：1000

任务要求

本任务中，你要实现一个含有三个输入的投票器。当输入 A,B,C 中，有不少于 2 个 1 时，你应该在唯一的输出 Y 中返回 1，否则返回 0。

部分功能表

$A + B + C$	Y
0	0
1	0
2	1
3	1

输入信号

输入	对应的输入信号编号
A	1
B	2
C	3

输出信号

输出	对应的输出信号编号
Y	4

评分方式

当你完全正确的实现本题时，可以获得 100% 的分数，否则获得 0% 的分数。

任务 2：译码器

最大允许门延迟：1000

题目描述

本任务中，你需要实现一个带使能 E 的 $5 - 32$ 译码器。其中，输入记为 $A[0 \dots 4]$ ，输出记为 $B[0 \dots 31]$ 。

部分功能表

下表给出带使能 E 的 $2 - 4$ 译码器的功能表（注意，可能与参考资料有所不同）：

E	$A[0]$	$A[1]$	$B[0]$	$B[1]$	$B[2]$	$B[3]$
0	X	X	0	0	0	0
1	0	0	1	0	0	0
1	1	0	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	1

根据上面的功能表，可以扩展到任意一个 $k - 2^k$ 译码器，本题等价于设计 $k = 5$ 时的译码器。

输入信号

输入	对应的输入信号编号
E	1
$A[0 \dots 4]$	2 ... 6

输出信号

输出	对应的输出信号编号
$B[0 \dots 31]$	7 ... 38

评分方式

下面规定 D 为 $A[0 \dots 4]$ 表示的二进制数。

对每个 $i \in \{2, 3, 4, 5\}$ ：在所有满足 $D < 2^i$ 的测试点中，在不考虑使能 E 的情况下正确输出结果可以获得 20% 的分数，考虑使能 E 可以获得额外 5% 的分数。

任务 3：选择器

最大允许门延迟：1000

题目描述

本任务中，你需要实现一个带使能 E 的 $5 - 32$ 选择器。其中，待查询地址记为 $A[0 \dots 4]$ ，信号集合记为 $D[0 \dots 31]$ ，输出记为 R 。

部分功能表

下表给出带使能 E 的 $2 - 4$ 选择器的部分功能表（注意，可能与参考资料有所不同）：

E	$A[0]$	$A[1]$	$D[0]$	$D[1]$	$D[2]$	$D[3]$	R
0	X	X	X	X	X	X	0
1	0	0	1	1	0	1	1
1	1	0	1	1	0	1	1
1	0	1	1	1	0	1	0
1	1	1	1	1	0	1	1

根据上面的功能表，可以扩展到任意一个 $k - 2^k$ 选择器，本题等价于设计 $k = 5$ 时的选择器。

输入信号

输入	对应的输入信号编号
E	1
$A[0 \dots 4]$	2 ... 6
$D[0 \dots 31]$	7 ... 38

输出信号

输出	对应的输出信号编号
R	39

评分方式

下面规定 D 为 $A[0 \dots 4]$ 表示的二进制数。

对每个 $i \in \{2, 3, 4, 5\}$ ：在所有满足 $D < 2^i$ 的测试点中，在不考虑使能 E 的情况下正确输出结果可以获得 20% 的分数，考虑使能 E 可以获得额外 5% 的分数。

任务 4：比较器

最大允许门延迟：1000

题目描述

本任务中，你需要实现一个 16 位的比较器，对输入的两个 16 位数字进行比较。其中，第一个数字为 $A[0 \dots 15]$ ，第二个数字为 $B[0 \dots 15]$ ，输出为 R 。输出信号为 1 当且仅当 A 代表的二进制数小于 B 代表的二进制数。

部分功能表

下表给出一个 2 位比较器的部分功能表：

$A[0]$	$A[1]$	$B[0]$	$B[1]$	R
0	0	0	1	1
1	0	0	1	1
0	1	0	1	0
1	1	0	1	0

根据上面的功能表，可以扩展到任意一个 k 位比较器。本题等价于设计 $k = 16$ 时的比较器。

输入信号

输入	对应的输入信号编号
$A[0 \dots 15]$	1 ... 16
$B[0 \dots 15]$	17 ... 32

输出信号

输出	对应的输出信号编号
R	33

评分方式

当你完全正确的实现本题时，可以获得 100% 的分数，否则获得 0% 的分数。

任务 5：加法器

最大允许门延迟：1000

题目描述

本任务中，你需要实现一个 16 位的加法器，对输入的两个 16 位数字进行加和，并舍去最高位的进位。其中，第一个数字为 $A[0 \dots 15]$ ，第二个数字为 $B[0 \dots 15]$ ，输出为 $Y[0 \dots 15]$ 。

部分功能表

下表给出一个 2 位比较器的部分功能表：

A[0]	A[1]	B[0]	B[1]	Y[0]	Y[1]	等式
0	0	0	0	0	0	$0 + 0 = 0$
1	0	0	1	1	1	$1 + 2 = 3$
1	0	1	0	0	1	$1 + 1 = 2$
1	1	0	1	1	0	$3 + 2 = 5$

根据上面的功能表，可以扩展到任意一个 k 位加法器。本题等价于设计 $k = 16$ 时的比较器。

输入信号

输入	对应的输入信号编号
$A[0 \dots 15]$	1 ... 16
$B[0 \dots 15]$	17 ... 32

输出信号

输出	对应的输出信号编号
$Y[0 \dots 15]$	33 ... 48

评分方式

对最大的 $i \in \{2, 3, 4, 5\}$ ，如果对于输入中所有满足除前 2^i 位都为 0 的数据均正确，即可得到 $(i - 1) \times 25\%$ 的分数。若没有满足的 i ，则得到 0% 的分数。

附加任务：超前进位加法器

最大允许门延迟：5

最多输入信号个数：5

题目描述

本任务为任务 5 的特殊任务。本题的实现需求和 **任务 5：加法器** 完全相同，但是添加了更严格的限制。在本题中，最大允许门延迟 $T_{comb} = 5$ ，同时最多输入信号个数 $MaxDegree = 5$ ，也就是每个逻辑门只能接受不超过 5 个输入信号。

输入信号

输入	对应的输入信号编号
A[0 ... 15]	1 ... 16
B[0 ... 15]	17 ... 32

输出信号

输出	对应的输出信号编号
Y[0 ... 15]	33 ... 48

评分方式

对于输入中所有满足除前四位都为 0 的数据均正确，即可得到 50% 的分数。当你完全正确的实现本题时，可以获得 100% 的分数。

注意：对答案的检验部分先于模拟部分。你的答案需要在符合本题限制的情况下，才可以获得分数。

任务 6：ALU

最大允许门延迟：1000

题目描述

本任务中，你需要设计一个包含 8 个操作的 ALU，对两个 16 位数字进行计算。其中，运算标识符为 $op[0 \dots 2]$ ，第一个数字为 $A[0 \dots 15]$ ，第二个数字为 $B[0 \dots 15]$ ，输出为 $Y[0 \dots 15]$ 。

下面为每个运算标识符对应的操作（表中使用数组名称代表对应的数字，同时对数字的运算都是位运算，可以参考学习手册的定义）：

$op[0]$	$op[1]$	$op[2]$	Y
0	0	0	$A + B$
1	0	0	$A - B$
0	1	0	$A \wedge B$
1	1	0	$A \vee B$
0	0	1	$A \oplus B$
1	0	1	$\neg A$
0	1	1	$A \ll (B \wedge 15)$
1	1	1	$A \gg (B \wedge 15)$

其中，左移结果应当舍去超过储存空间的位，也就是溢出。

输入信号

输入	对应的输入信号编号
$op[0 \dots 2]$	1 ... 3
$A[0 \dots 15]$	4 ... 19
$B[0 \dots 15]$	20 ... 35

输出信号

输出	对应的输出信号编号
$Y[0 \dots 15]$	36 ... 51

评分方式

正确实现 8 种运算的 i 种，将会得到 $i \times 12.5\%$ 的分数。

时序逻辑电路部分

在这一部分中，你需要利用理想状态下的时序逻辑电路，完成一些复杂的元件设计需求，并最终实现一个简易的 CPU。

下面是一些约定：

- 在未说明的情况下，最大允许门延迟 T_{comb} 为 1000。
- 在未说明的情况下，每个多输入逻辑门可以接受的最多输入信号个数 $MaxDegree$ 为 10。
- 在未说明的情况下，可使用的正边沿 D-触发器个数 $MaxDFF$ 为 1000。
- 在输出中，不应当包含 x 状态。你可以利用一些操作消除 x 带来的影响，详见学习手册。
- 输入中涉及到的数字和指令在输入信号中均以二进制从低到高的顺序给出。
- 功能表中的 x 和学习手册定义的特殊数字信号不同，代表的是 **取 0 或者 1 都成立** 的通配符。
- 在下面的说明中，**时钟信号正跳变前** 指代的是当前周期的最开始，**时钟信号正跳变后** 指代的是当前周期的结束。
- 使用 0 代表将指令中对应部分全部用 0 填充。
- 所有正边沿 D-触发器的默认储存值均视为 x。
- 更宏观的约定已经在学习手册中的 **答案格式** 中给出。

任务 1：串行奇偶校验器

最大允许门延迟：1000

题目描述

本任务中，你需要实现一个串行奇偶校验器。其中，输入为 F 和 X，输出为 Y。

假设共运行 n 个周期。对于一个固定的数列 x_1, \dots, x_n ，在第 $i \in [1, n]$ 次时钟信号正跳变前，输入信号 X 的值会被设置为 x_i ，而在正跳变后，应当输出 $Y = \bigoplus_{j=1}^i x_j$ ，其中 \oplus 为异或。

为了方便实现，增设输入 F，代表是否处于第一个周期。F 为 1 当且仅当目前处于第一个周期。

输入信号

输入	对应的输入信号编号
F	1
X	2

输出信号

输出	对应的输出信号编号
Y	3

评分方式

当你完全正确的实现本题时，可以获得 100% 的分数，否则获得 0% 的分数。

任务 2：移位寄存器

最大允许门延迟：3

D_FF 个数限制：16

题目描述

本任务中，你需要实现一个移位寄存器。其中，输入为 F 和 $X[0 \dots 15]$ ，输出为 Y 。 F 的定义与上一题中的定义相同，不再赘述。

本题共运行 16 个周期，对第 $i \in [1, 16]$ 个周期，在正跳变后，应当输出 $Y = X[i - 1]$ 。 $X[0 \dots 15]$ 仅在第一个周期中提供，也即在其他周期中，输入将全都是 0。

请留意本题特殊的 D_FF 个数限制。

输入信号

输入	对应的输入信号编号
F	1
$X[0 \dots 15]$	2 ... 17

输出信号

输出	对应的输出信号编号
Y	18

评分方式

当你完全正确的实现本题时，可以获得 100% 的分数，否则获得 0% 的分数。

任务 3：可变模计数器

最大允许门延迟：1000

题目描述

本任务中，你需要实现一个可变模计数器。其中，输入为 RST 和 $M[0 \dots 3]$ ，输出为 $CNT[0 \dots 3]$ 。

本任务需要维护一个变量 CNT 。在每个周期，如果 RST 为 1，则需要将 CNT 重置为 0。否则，需要令 $CNT \leftarrow (CNT + 1) \bmod (M + 1)$ 。每个周期正跳变后应当输出 CNT 的目前值。

本题中的模数系数 M 满足 $M > 0$ ，并且仅会在 RST 为 1 时发生变化。 M 的值在每个周期中都会提供。保证第一个周期中 RST 为 1。

输入信号

输入	对应的输入信号编号
RST	1
$M[0 \dots 3]$	2 ... 5

输出信号

输出	对应的输出信号编号
$CNT[0 \dots 3]$	6 ... 9

评分方式

当你完全正确的实现本题时，可以获得 100% 的分数，否则获得 0% 的分数。

任务 4：乘法器

最大允许门延迟：10

D_FF 个数限制：50

注意：以上限制为获得全部分数需要满足的限制，其余部分分将会在下面给出。

题目描述

本任务中，你需要实现一个 16 位的乘法器。其中，第一个数字为 $A[0 \dots 15]$ ，第二个数字为 $B[0 \dots 15]$ ，输出为 $Y[0 \dots 31]$ 。需要注意的是，此时的储存空间是足够的，并不存在溢出的情况。

另外，本任务要求乘法器在尽可能高的效率下工作，因此推荐先完成组合逻辑部分中的**超前进位加法器**。乘法器将会运行 20 个周期，而只需要在任意一个周期输出正确的结果，就会判作正确。

为了方便实现，增设输入 F，代表是否处于第一个周期。同时，A 和 B 仅在第一个周期中提供，也即在其他周期中，输入将全都是 0。

输入信号

输入	对应的输入信号编号
$A[0 \dots 15]$	1 ... 16
$B[0 \dots 15]$	17 ... 32
F	33

输出信号

输出	对应的输出信号编号
$Y[0 \dots 31]$	34 ... 65

评分方式

在成功实现的情况下，如果超出门延迟限制，可以获得 50% 的分数。

在成功实现且未超出门延迟限制的情况下，如果 D_FF 个数在 51 和 66 之间，可以获得 75% 的分数；如果 D_FF 个数在 50 个及以下，可以获得 100% 的分数。

任务 5：寄存器堆

最大允许门延迟：1000

题目描述

本任务中，你需要实现一个寄存器堆，对 32 个 16 位整数的寄存器进行维护。输入包含 RST 和 $CMD[0 \dots 31]$ ，输出包含 $RDA[0 \dots 15]$ 和 $RDB[0 \dots 15]$ 。

在寄存器堆架构中，正如每道题目保证的一样，地址为 0 的寄存器保持为 0 并不支持写入操作。在 RST 为 1 时，所有 32 个寄存器应当全部清空。否则， $CMD[0 \dots 31]$ 将会包含一个指令，描述本次操作的行为。下面介绍指令的结构。

31	16 15	11 10	10 9	5 4	0
wd[15 ... 0]		rd	We	rs2	rs1

我们将指令从高位到低位，也就是输入信号顺序从后到前排列一个指令。需要注意的是，参数的顺序也进行了交换，例如图中 wd[15 ... 0]，在实际的输入中顺序应当是 wd[0 ... 15]。

由于大多数寄存器读取都需要读出两个值进行计算，因此在指令的开始给出了 rs1 和 rs2 两个地址，分别代表需要读取的信息地址。你应当在每个周期中输出这两个地址对应的值，分别对应 $RDA[0 \dots 15]$ 和 $RDB[0 \dots 15]$ 。

一个指令可能存在写操作。指令中 We 提供写使能，只有 We 为 1 时才进行写操作。随后的 rd 和 wd[15 ... 0] 给出了写入的地址和写入的信息，保证 $rd \neq 0$ 。需要注意，指令运行需要满足 **先写后读**，也就是先向寄存器堆进行必要的写入，再从中读取两个值。

保证第一个操作为清空操作，且清空操作下指令输入全为 0。保证在 $We = 0$ 时 rd 和 wd 都是 0。

输入信号

输入	对应的输入信号编号
RST	1
$CMD[0 \dots 31]$	2 ... 33

输出信号

输出	对应的输出信号编号
$RDA[0 \dots 15]$	34 ... 49
$RDB[0 \dots 15]$	50 ... 65

评分方式

当你完全正确的实现本题时，可以获得 100% 的分数，否则获得 0% 的分数。

任务 6：奋斗四小时，手搓 CPU

最大允许门延迟：1000

题目描述

本任务中，你需要实现一个简易 CPU，包含一个寄存器堆，一个 ALU，以及其余零散的部件。输入包含 RST 和 CMD[0 ... 31]，输出包含 RD[0 ... 15]。

在 RST 为 1 时，所有 32 个寄存器应当全部清空。否则，CMD[0 ... 31] 将会包含一个指令，描述本次操作的行为。下面介绍指令的结构。

本任务的指令从 RISC-V 标准中修改，共有 3 种指令，分别为 R-type，I-type POKE 和 I-type PEEK。

R-type

31	25 24	20 19	15 14	12 11	7 6	0
0	rs2	rs1	funct3	rd	0010011	

R-type 指令描述了一个 ALU 操作，其中 ALU 的架构和组合逻辑电路部分的 ALU 相同。指令中，rs1 和 rs2 分别确定左操作数和右操作数的地址，funct3 指明操作类型，在计算完后将结果储存到 rd 对应的寄存器中，并保证 $rd \neq 0$ 。在周期结束时，输出应当全为 0。

I-type POKE

31 31 30	15 14	12 11	7 6	0
0	imm[15 ... 0]	0	rd	0100011

I-type POKE 指令描述了一个写入操作，其中 imm[15 ... 0] 描述了一个立即数，表示需要储存到寄存器中的信息，而 rd 指定储存地址，并保证 $rd \neq 0$ 。在周期结束时，输出应当全为 0。

I-type PEEK

31	15 14	12 11	7 6	0
0	0	rs	0000011	

I-type PEEK 指令描述了一个读取操作，其中 rs 指定了需要读取的寄存器地址。在周期结束时，这个指令需要输出 rs 地址对应的寄存器值。

保证第一个操作为清空操作，且清空操作下指令输入全为 0。

输入信号

输入	对应的输入信号编号
RST	1
CMD[0 ... 31]	2 ... 33

输出信号

输出	对应的输出信号编号
$RD[0 \dots 15]$	34 ... 49

评分方式

当你完全正确的实现本题时，可以获得 100% 的分数，否则获得 0% 的分数。