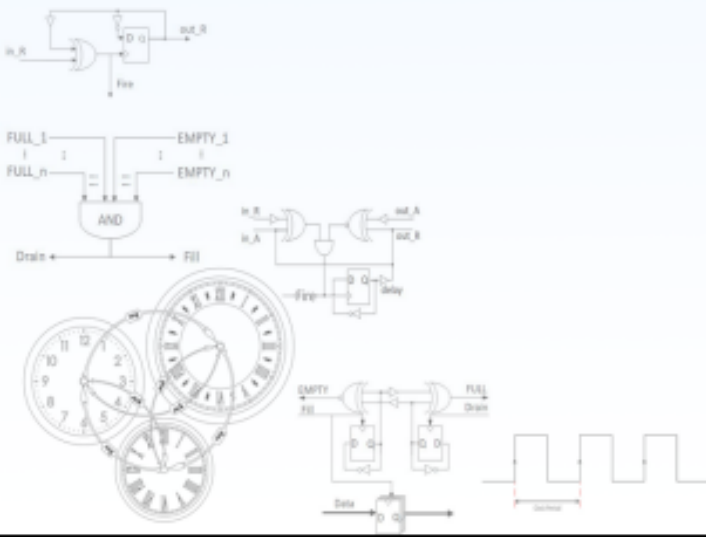
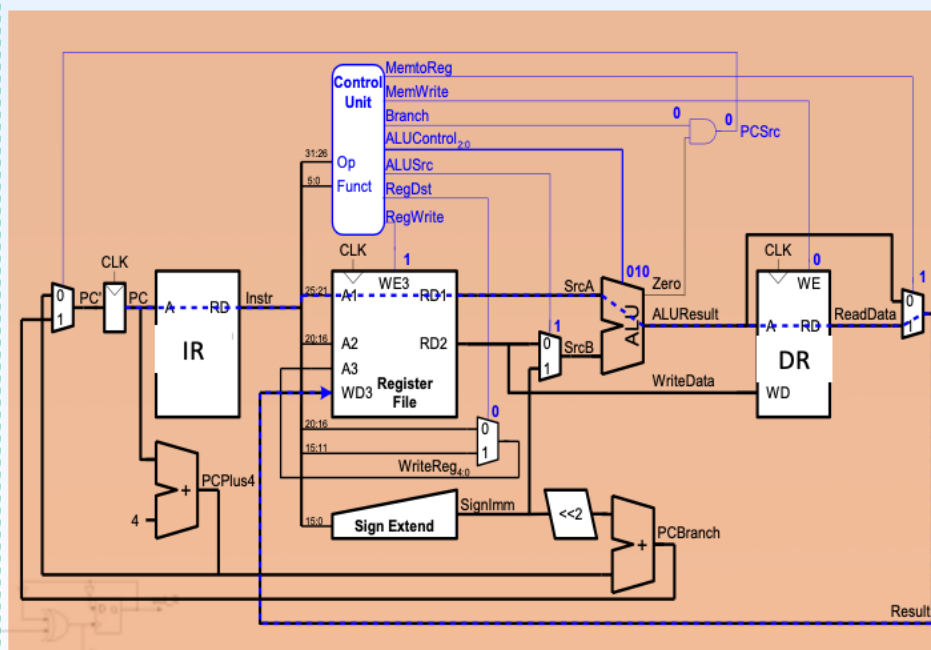


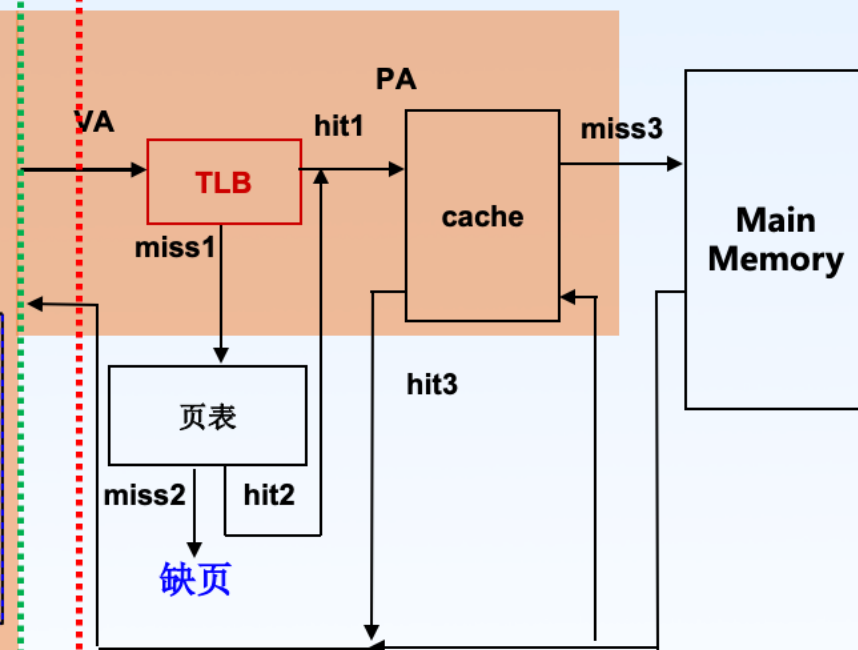


指令、CPU和层次化存储 作业和课程设计





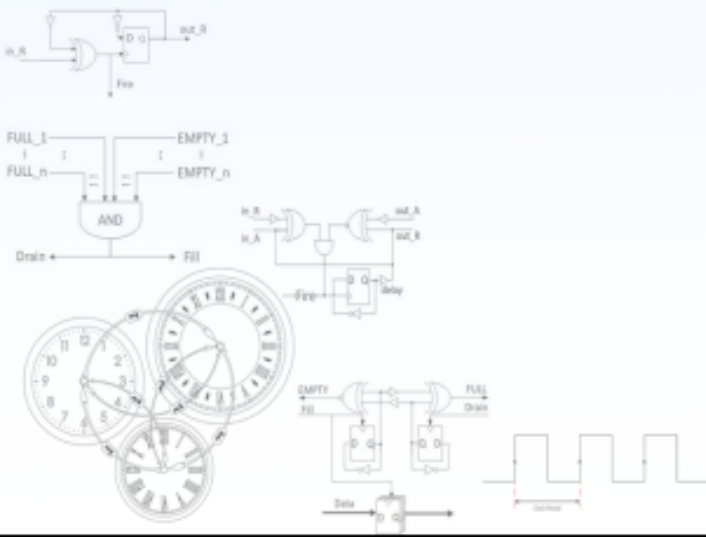
CPU核



存储系统



作业



高级语言到汇编程序——编译

```
sum = 0;
for (i = 0; i < 2; i++)
    sum += a[i];
*v = sum;
```

高级语言

汇编语言

```
l0:      sum <-- 0
l1:      ap <-- A  A是数组a的起始地址
l2:      i  <-- 0
l3:      if (i >= 2) goto done
l4:  loop: t  <-- (ap) 数组元素a[i]的值
l5:      sum <-- sum + t  累计在sum中
l6:      ap <-- ap + 4  计算下个数组元素地址
l7:      i  <-- i + 1
l8:      if (i < n) goto loop
l9:  done: V <-- sum  累计结果保存至地址v
```

汇编程序到机器语言——汇编

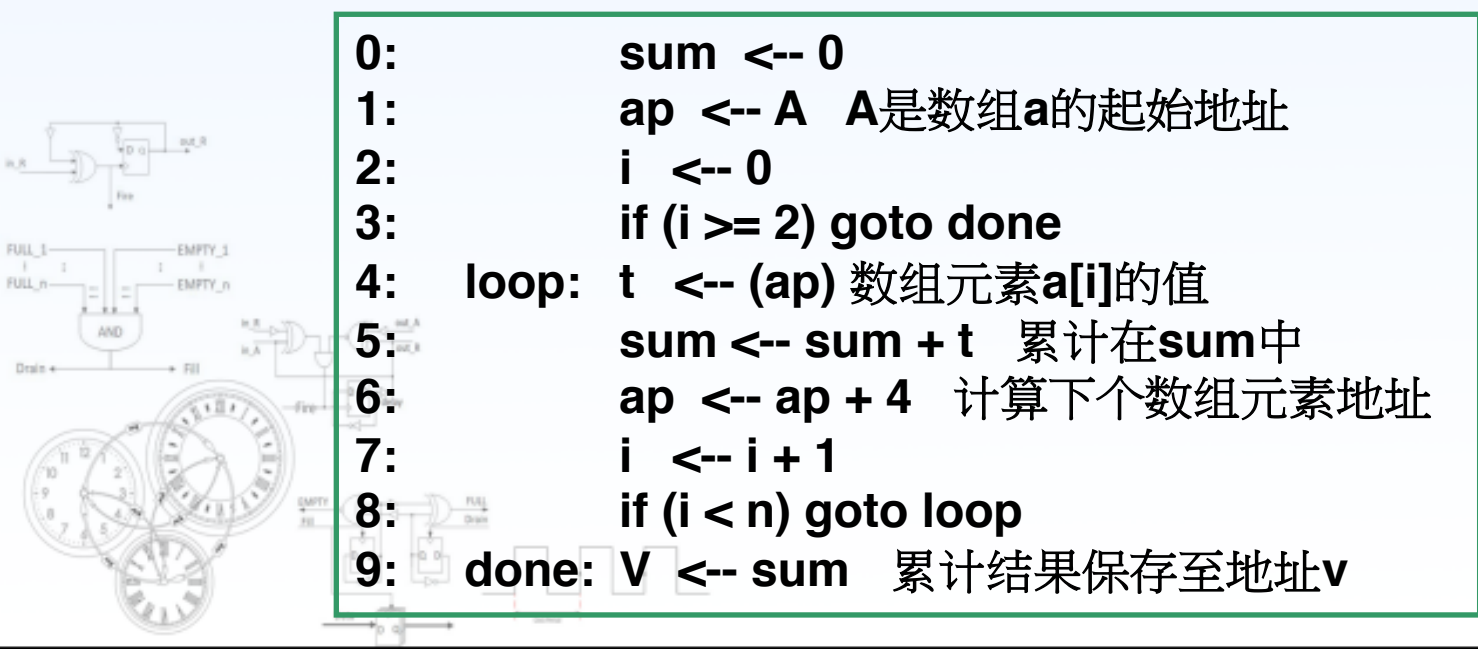
作业1: 请使用介绍过的32位MIPS指令，将框图中的汇编语言的第1行、第3行、第6行和9行”的汇编语句，手工翻译成MIPS指令

■ 32个寄存器功能可参考：

<https://www.cnblogs.com/s08243/p/7693387.html>

■ MIPS指令建议使用已经在课堂上分析过的：包括：ADDU、SUBU、ORI、LW、SW、BEQ、J

■ （在操作系统空间、程序空间，虚拟空间中）数组a的起始地址A为0x00002300，地址V的值为0x0030F1BC



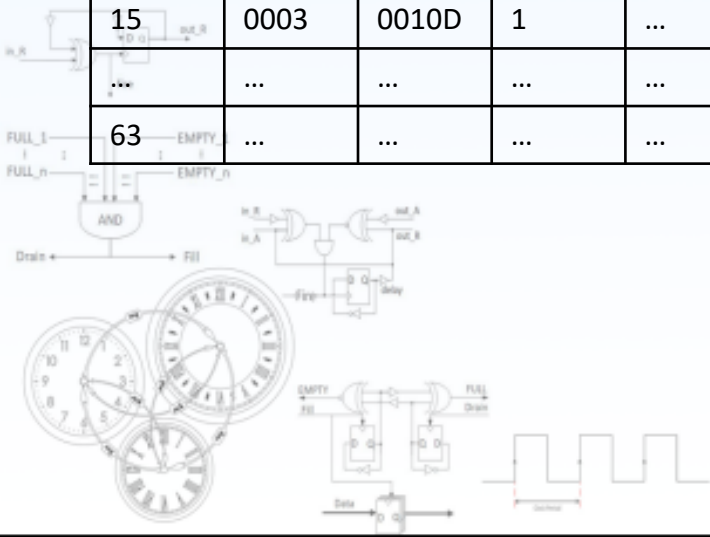
```
0:      sum <-- 0
1:      ap <-- A  A是数组a的起始地址
2:      i  <-- 0
3:      if (i >= 2) goto done
4:  loop: t  <-- (ap) 数组元素a[i]的值
5:      sum <-- sum + t  累计在sum中
6:      ap <-- ap + 4  计算下个数组元素地址
7:      i  <-- i + 1
8:      if (i < n) goto loop
9:  done: V <-- sum  累计结果保存至地址v
```

分页管理 (1)

●32位系统采用分页管理地址，描述如下：

- 虚拟地址和物理地址均为32位，每一页的大小为4KB
- 快表共256行，采用4路组相联映射，快表如下图，页表见下页。

行索引	第一组			第二组			第三组			第四组		
	tag	实页号	有效位	tag	实页号	有效位	tag	实页号	有效位	tag	实页号	有效位
0	0003	-	0	0009	000FD	1	0000	-	0	0700	00021	1
1	0003	1002D	1	0108	-	0	0004	-	0	A001	-	0
2	F002	-	0	0002	00033	0	12F6	-	0	0033	-	0
3	0007	-	0	0003	00F0D	1	0AAA	3BF04	1	F022	-	0
...
15	0003	0010D	1
...
63



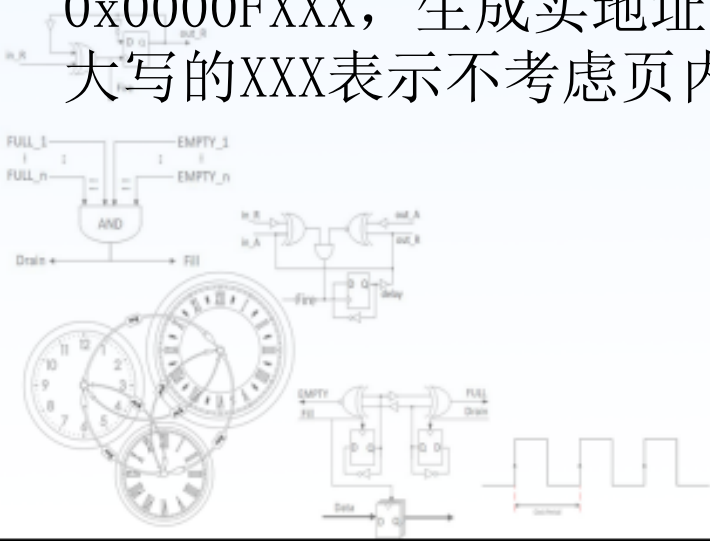
分页管理 (2)

●32位系统采用分页管理地址，描述如下：

- 虚拟地址和物理地址均为32位，每一页的大小为4KB
- 快表共256行，采用4路组相联映射，快表见上页，页表见右图

作业2： 请根据分页管理的机制，描述虚地址0x00002XXX、0x0030FXXX和0x0000FXXX，生成实地址的过程（这里大写的XXX表示不考虑页内地址）。

虚页号	实页号	有效位
00000	00028	1
00001	-	0
00002	00033	1
00003	00002	1
00004	-	0
...
0000E	0002D	1
0000F	-	0
...
0030F	0010D	1
...



Cache

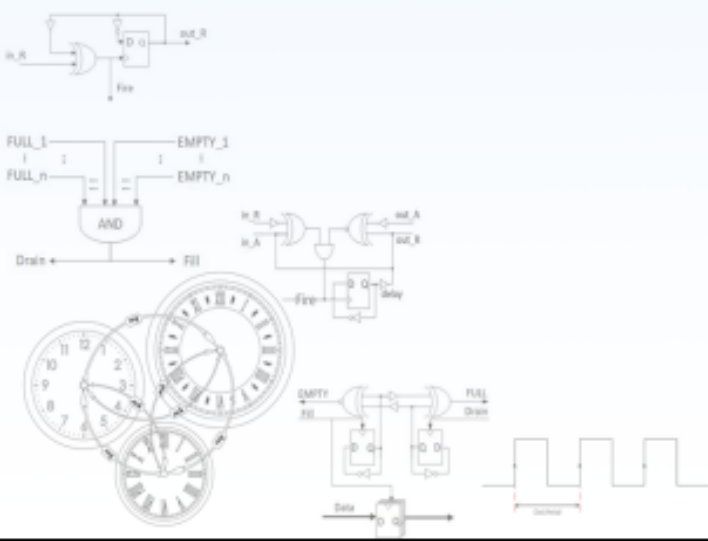
- 32位系统cache数据区为64KB，每一槽（块）的大小为16B，采用4路组相联映射。一次取32位数据。部分cache如下图。

作业3：请分析物理地址0x0010D1BC，0x00033000和0x00000004所对应的32位数据。

id x	第一组						第二组						第三组						第四组					
	tag	v	D0	D1	D2	D3	tag	v	D0	D1	D2	D3	tag	v	D0	D1	D2	D3	tag	v	D0	D1	D2	D3
0	0000	0	F1	BC	1C	AB	0000	1A	AB	0C	A0	00	00D0	1	00	00	00	10	0030	0	0F	FF	FF	FD
1																								
...
768	A011	1	AB	1D	31	A1	0003	1	0B	0A	00	00	0011	1	00	10	00	AA	01D0	1	0B	0C	00	01
...
3355	0000	0	AA	CA	0D	F0	010C	1	1A	0B	11	1C	0033	0	CC	AB	0A	00	0010	1	00	01	00	01
...
4096

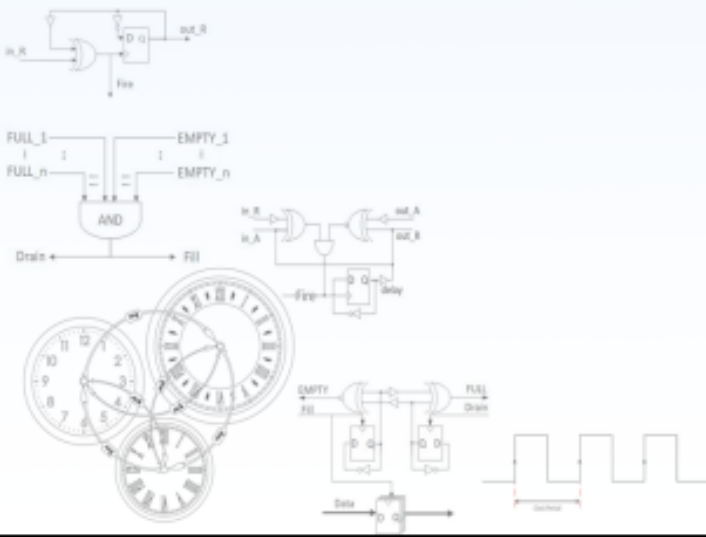
CPU执行过程

作业4: 请分析作业1中MIPS指令的执行过程（参考作业2和作业3）。





课程设计



请设计一个CPU，要求

- 可以执行简单的MIPS指令，至少包括：
 - ADDU、SUBU、ORI、LW、SW、BEQ、J
- 采用单周期/多周期/流水线方式均可
- 要求采用vivado前仿（功能仿真）成功
- 作业提交形式

- 报告

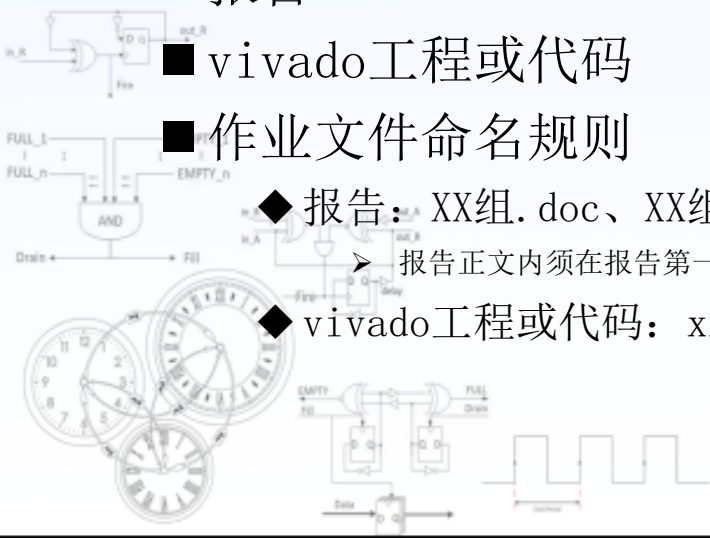
- vivado工程或代码

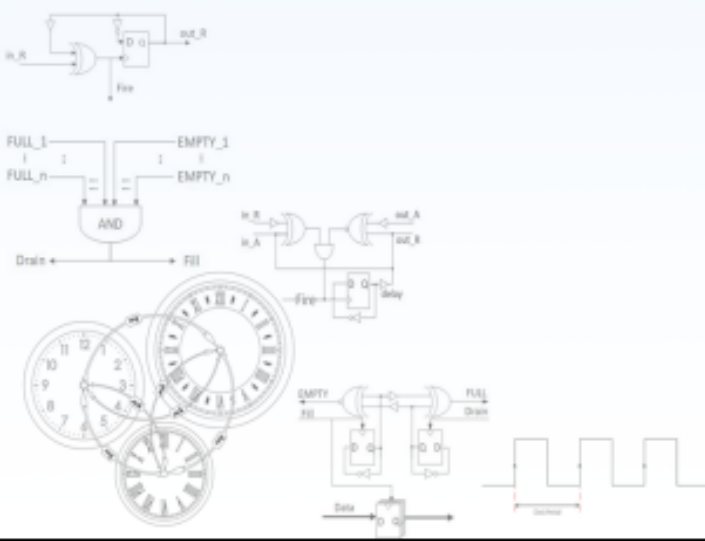
- 作业文件命名规则

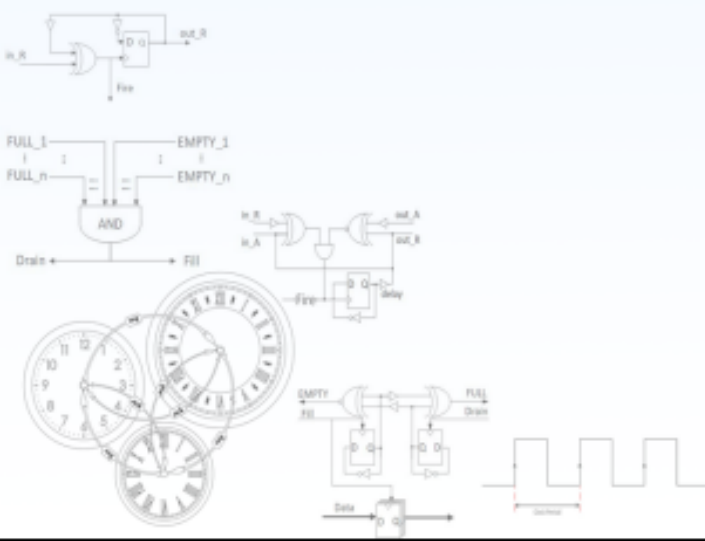
- ◆报告：XX组.doc、XX组.docx、XX组.pdf

- 报告正文内须在报告第一页注明组员姓名和班级

- ◆vivado工程或代码：xx组.tar、xx组.zip等







汇编程序到机器语言——汇编

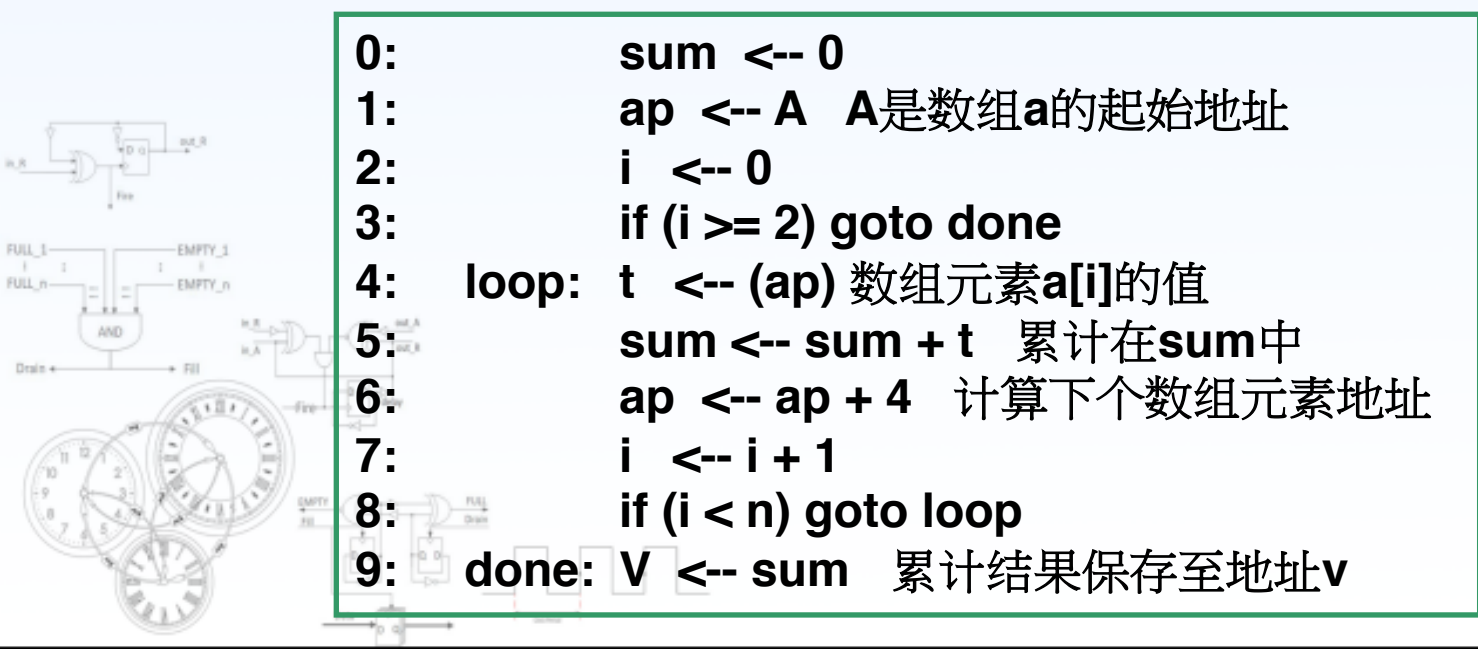
作业1: 请使用介绍过的32位MIPS指令，将框图中的汇编语言的第1行、第3行、第6行和9行”的汇编语句，手工翻译成MIPS指令

■ 32个寄存器功能可参考：

<https://www.cnblogs.com/s08243/p/7693387.html>

■ MIPS指令建议使用已经在课堂上分析过的：包括：ADDU、SUBU、ORI、LW、SW、BEQ、J

■ （在操作系统空间、程序空间，虚拟空间中）数组a的起始地址A为0x00002，地址V的值为0x0030F



```
0:      sum <-- 0
1:      ap <-- A  A是数组a的起始地址
2:      i  <-- 0
3:      if (i >= 2) goto done
4:  loop: t  <-- (ap) 数组元素a[i]的值
5:      sum <-- sum + t  累计在sum中
6:      ap <-- ap + 4  计算下个数组元素地址
7:      i  <-- i + 1
8:      if (i < n) goto loop
9:  done: V <-- sum  累计结果保存至地址v
```

作业2答案

作业2: 请根据分页管理的机制，描述虚地址0x00002XXX、0x0030FXXX和0x0000FXXX，生成实地址的过程（这里大写的XXX，代表页内地址未知）。

tag0000-line02-offsetXXX
tag0003-line0F-offsetXXX
tag0000-line0F-offsetXXX

虚页号	实页号	有效位
00000	00028	1
00001	-	0
00002	00033	1
00003	00002	1
00004	-	0
...
0000E	0002D	1
0000F	-	0
...
0030F	0010D	1
...

行索引	第一组			第二组			第三组			第四组		
	tag	实页号	有效位	tag	实页号	有效位	tag	实页号	有效位	tag	实页号	有效位
0	0003	-	0	0009	000FD	1	0000	-	0	0700	00021	1
1	0003	1002D	1	0108	-	0	0004	-	0	A001	-	0
2	F002	-	0	0002	00033	0	12F6	-	0	0033	-	0
3	0007	-	0	0003	00F0D	1	0AAA	3BF04	1	F022	-	0
...
15	0003	0010D	1	0010	00ABC	0	0101	3BF0A	1	0001	0AABB	1
...
63

Cache

● 作业3:

■ 0x0010D1BC

■ 0x00033000

id x	第一组						第二组						第三组						第四组					
	tag	v	D0	D1	D2	D3	tag	v	D0	D1	D2	D3	tag	v	D0	D1	D2	D3	tag	v	D0	D1	D2	D3
0	00 00	0	F1	BC	1C	AB	00 00	1A	AB	0C	A0	00	00 D0	1	00	00	00	10	00 30	0	0F	FF	FF	FD
1																								
...
76 8	A0 11	1	AB	1D	31	A1	00 03	1	0B	0A	00	00	00 11	1	00	10	00	AA	01 D0	1	0B	0C	00	01
...
33 55	00 00	0	AA	CA	0D	F0	01 0C	1	1A	0B	11	1C	00 33	0	CC	AB	0A	00	00 10	1	00	01	00	01
...
40 96



问题和讨论

