# VE370 RC8

# Lecture Slides

# Exercise 5.3.5

There are many different design parameters that are important to a cache's overall performance. The table below lists parameters for different direct-mapped cache designs.

| | Cache Data Size | Cache Block Size | Cache Access Time |
|---|---|---|---|
| **a.** | 32 KB | 2 words | 1 cycle |
| **b.** | 32 KB | 4 words | 2 cycle |

**5.3.5** [20] <5.2, 5.3> Generate a series of read requests that have a lower miss rate on a 2 KB 2-way set associative cache than the cache listed in the table. Identify one possible solution that would make the cache listed in the table have an equal or lower miss rate than the 2 KB cache. Discuss the advantages and disadvantages of such a solution.

To have a lower miss rate on a 2KB 2-way set associative cache:
Eg. 2KB cache, 16-word per block, 2-set
word address: 0, 64, 0, 64, ….
0 and 64 are in the same set.

For a larger direct-mapped cache to have a lower or equal miss rate than a smaller 2-way set associative cache, it would need to have at least **double** the cache block size. The advantage of such a solution is less misses for near by addresses (spatial locality), but with the disadvantage of suffering longer access times.

# Exercise 5.5.2

**Exercise 5.5**

Recall that we have two write policies and write allocation policies, and their combinations can be implemented either in L1 or L2 cache.

| | L1 | L2 |
|---|---|---|
| **a.** | Write through, non-write allocate | Write back, write allocate |
| **b.** | Write through, write allocate | Write back, write allocate |

**5.5.2** [20] <5.2, 5.5> Describe the procedure of handling an L1 write-miss, considering the component involved and the possibility of replacing a dirty block.

a.

If L1 miss, send write request to L2.

If L2 hit, write data to L2, set the dirty bit.

If L2 miss, allocate cache block for the missing data, select a replacement victim.

If victim dirty, put it into the write-back buffer, which will be further forwarded into memory.

Issue write miss request to the memory.

Data arrives and is installed in L2 cache.

Write data to L2, set the dirty bit.

Processor resumes execution.

# Exercise 5.5.2

**Exercise 5.5**

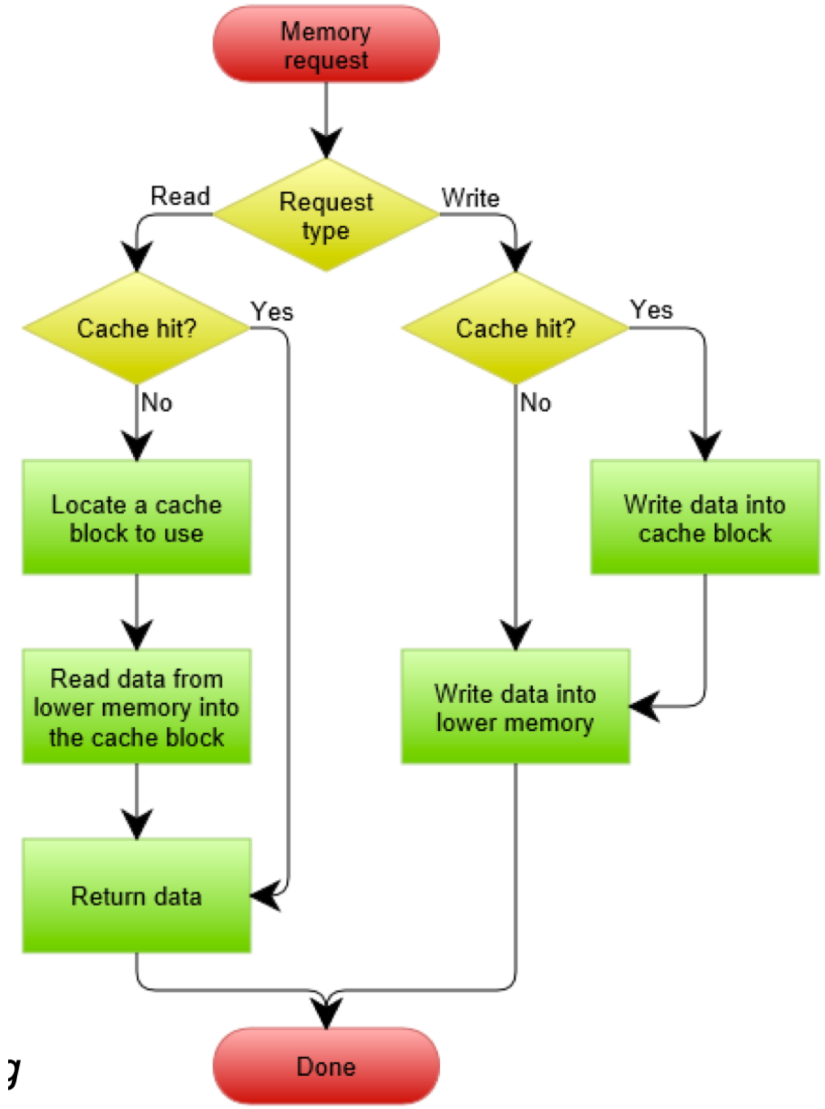
Recall that we have two write policies and write allocation policies, and their combinations can be implemented either in L1 or L2 cache.

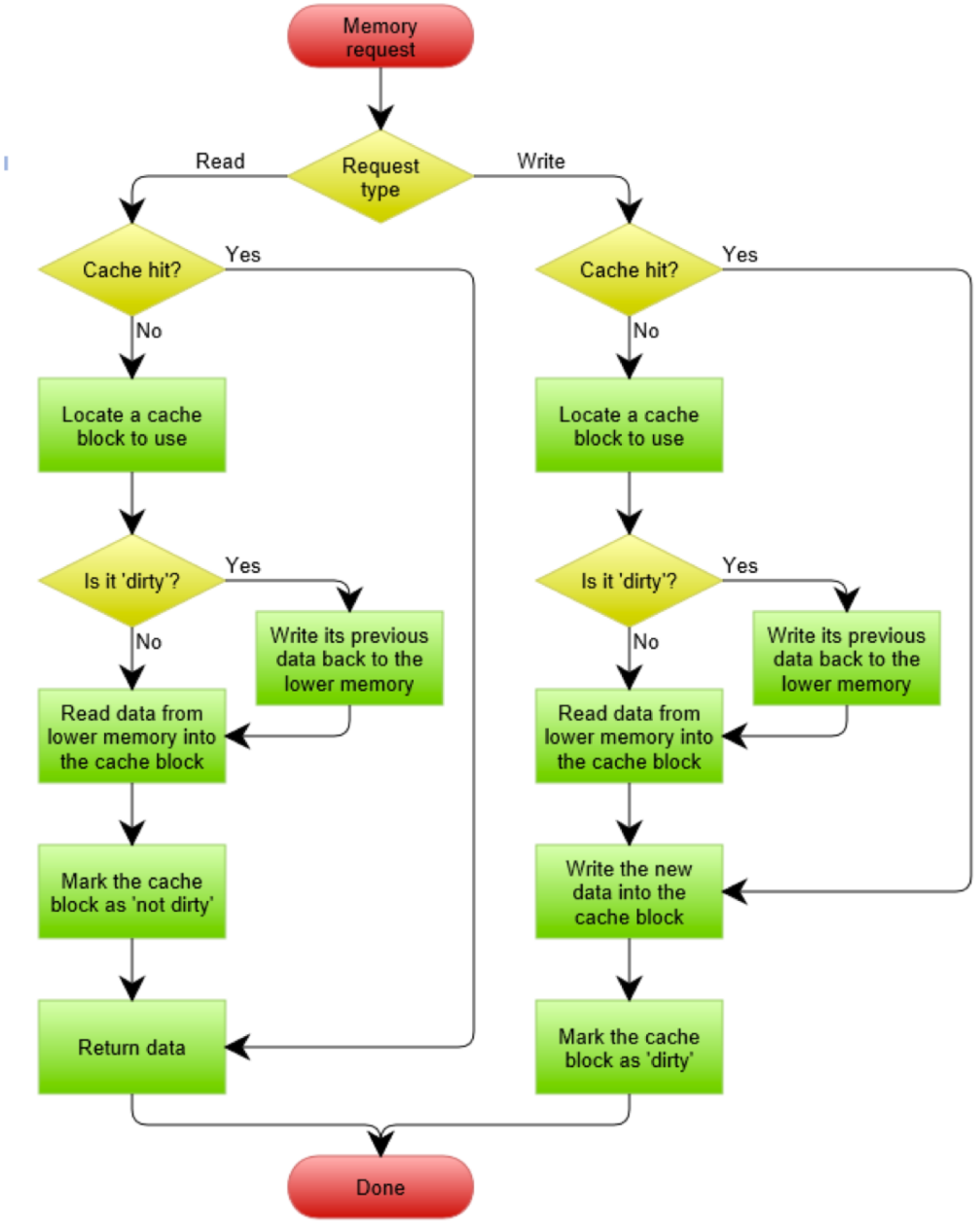
| | L1 | L2 |
|---|---|---|
| **a.** | Write through, non-write allocate | Write back, write allocate |
| **b.** | Write through, write allocate | Write back, write allocate |

**5.5.2** [20] <5.2, 5.5> Describe the procedure of handling an L1 write-miss, considering the component involved and the possibility of replacing a dirty block.

b.

1. If L1 miss, send write request to L2.
2. If L2 hit, write data to L2, set the dirty bit. (go to step 8)
3. If L2 miss, allocate cache block for the missing data, select a replacement victim.
4. If victim dirty, put it into the write-back buffer, which will be further forwarded into memory.
5. Issue write miss request to the memory.
6. Data arrives and is installed in L2 cache.
7. Write data to L2, set the dirty bit.
8. Data arrives and is installed in L1 cache.
9. Processor resumes execution and hits in L1 cache


Memory request
Request type
Read
Write
Cache hit?
Yes
No
Locate a cache block to use
Read data from lower memory into the cache block
Return data
Cache hit?
Yes
No
Write data into cache block
Write data into lower memory
Done


Memory request
Request type
Read
Write
Cache hit?
Yes
No
Locate a cache block to use
Is it 'dirty'?
Yes
No
Write its previous data back to the lower memory
Read data from lower memory into the cache block
Mark the cache block as 'not dirty'
Return data
Cache hit?
Yes
No
Locate a cache block to use
Is it 'dirty'?
Yes
No
Write its previous data back to the lower memory
Read data from lower memory into the cache block
Write the new data into the cache block
Mark the cache block as 'dirty'
Done

# Exercise 5.5.5

Consider the following program and cache behaviors.

| | Data Reads per 1000 Instructions | Data Writes per 1000 Instructions | Instruction Cache Miss Rate | Data Cache Miss Rate | Block Size (byte) |
|---|---|---|---|---|---|
| **a.** | 250 | 100 | 0.30% | 2% | 64 |
| **b.** | 200 | 100 | 0.30% | 2% | 64 |

**5.5.5** [5] <5.2, 5.5> For a write-back, write-allocate cache, assuming 30% of replaced data cache blocks are dirty, what are the minimal read and write bandwidths needed for a CPI of 2?

With the write back policy, the cache content may be changed and inconsistent with that in the main memory. Thus, upon a read miss, if the replaced block is “dirty”, it will first be written back to the main memory, and then the desired block will be brought into the cache. With write allocation, when a write miss occurs, the corresponding block will be brought to the cache, and then new data will be written into this block. The replaced block, if the dirty bit is set, will need to be written into the main memory first before the write block is brought into the cache. Suppose number of instruction is I, band width W, base CPI=1

a.

Read miss penalty: $I \times 0.25 \times 0.02 \times (1 + 0.3) \times (\frac{64}{W})$

Write miss penalty: $I \times 0.1 \times 0.02 \times (1 + 0.3) \times (\frac{64}{W})$

Instruction miss penalty: $I \times 0.003 \times (\frac{64}{W})$

$$I + I \times 0.25 \times 0.02 \times (1 + 0.3) \times \left(\frac{64}{W}\right) + I \times 0.1 \times 0.02 \times (1 + 0.3) \times \left(\frac{64}{W}\right) + I \times 0.003 \times (\frac{64}{W}) \leq 2I$$

# Exercise 5.7

## Exercise 5.7

In this exercise, we will look at the different ways capacity affects overall performance. In general, cache access time is proportional to capacity. Assume that main memory accesses take 70 ns and that memory accesses are 36% of all instructions. The following table shows data for L1 caches attached to each of two processors, P1 and P2.

| | | L1 Size | L1 Miss Rate | L1 Hit Time |
|---|---|---|---|---|
| **a.** | P1 | 2 KB | 8.0% | 0.66 ns |
| | P2 | 4 KB | 6.0% | 0.90 ns |
| **b.** | P1 | 16 KB | 3.4% | 1.08 ns |
| | P2 | 32 KB | 2.9% | 2.02 ns |

**5.7.1** [5] <5.3> Assuming that the L1 hit time determines the cycle times for P1 and P2, what are their respective clock rates?

**5.7.3** [5] <5.3> Assuming a base CPI of 1.0 without any memory stalls, what is the total CPI for P1 and P2? Which processor is faster?

5.7.1

$$Clock\ rate = \frac{1}{hit\ time}$$

a. P1: 1.51GHz P2: 1.11GHz

# Exercise 5.7.3

**5.7.3** [5] <5.3> Assuming a base CPI of 1.0 without any memory stalls, what is the total CPI for P1 and P2? Which processor is faster?

5.7.3

a.

first calculate AMAT for P1 &P2

P1: $0.08 * 70 + 1 * 0.66 = 6.26ns$, using $\frac{6.26}{0.66} = 9.48$ clock cycles;

P2 similarly. Answer is 5.1ns, using 5.67 clock cycles.

Second, P1:

$$9.48\times0.36 + 1\times0.64 = 4.05\ clock\ cycles.$$
$$4.05\times0.66(ns/clock) = 2.673\ ns$$

P2:

$$5.67\times0.36 + 1\times0.64 = 2.68\ clock\ cycles$$
$$2.68\times0.9(ns/clock) = 2.413\ ns$$

# Exercise 5.7.4-5.7.6

For the next three problems, we will consider the addition of an L2 cache to P1 to presumably make up for its limited L1 cache capacity. Use the L1 cache capacities and hit times from the previous table when solving these problems. The L2 miss rate indicated is its local miss rate.

| | L2 Size | L2 Miss Rate | L2 Hit Time |
|---|---|---|---|
| **a.** | 1 MB | 95% | 5.62 ns |
| **b.** | 8 MB | 68% | 23.52 ns |

**5.7.4** [10] <5.3> What is the AMAT for P1 with the addition of an L2 cache? Is the AMAT better or worse with the L2 cache?

**5.7.5** [5] <5.3> Assuming a base CPI of 1.0 without any memory stalls, what is the total CPI for P1 with the addition of an L2 cache?

**5.7.6** [10] <5.3> Which processor is faster, now that P1 has an L2 cache? If P1 is faster, what miss rate would P2 need in its L1 cache to match P1's performance? If P2 is faster, what miss rate would P1 need in its L1 cache to match P2's performance?

5.7.4

a.

first calculate AMAT for P1 &P2

P1:

$$1\times0.66 + 0.08\times5.62 + 0.08\times0.95\times70 = 6.43ns$$

$$\frac{6.43}{0.66} = 9.74 > 9.48$$

Worse

P2:

$$1\times0.9 + 0.06\times5.62 + 0.06\times0.95\times70 = 5.22ns$$

$$\frac{5.22}{0.9} = 5.808 > 5.67$$

worse

# Exercise 5.7.4-5.7.6

For the next three problems, we will consider the addition of an L2 cache to P1 to presumably make up for its limited L1 cache capacity. Use the L1 cache capacities and hit times from the previous table when solving these problems. The L2 miss rate indicated is its local miss rate.

| | L2 Size | L2 Miss Rate | L2 Hit Time |
|---|---|---|---|
| **a.** | 1 MB | 95% | 5.62 ns |
| **b.** | 8 MB | 68% | 23.52 ns |

**5.7.4** [10] <5.3> What is the AMAT for P1 with the addition of an L2 cache? Is the AMAT better or worse with the L2 cache?

**5.7.5** [5] <5.3> Assuming a base CPI of 1.0 without any memory stalls, what is the total CPI for P1 with the addition of an L2 cache?

**5.7.6** [10] <5.3> Which processor is faster, now that P1 has an L2 cache? If P1 is faster, what miss rate would P2 need in its L1 cache to match P1's performance? If P2 is faster, what miss rate would P1 need in its L1 cache to match P2's performance?

5.7.5

a.

P1:

$$9.74 \times 0.36 + 1 \times 0.64 = 4.15$$

P2:

$$5.808 \times 0.36 + 1 \times 0.64 = 2.73$$

# Exercise 5.7.4-5.7.6

For the next three problems, we will consider the addition of an L2 cache to P1 to presumably make up for its limited L1 cache capacity. Use the L1 cache capacities and hit times from the previous table when solving these problems. The L2 miss rate indicated is its local miss rate.

| | L2 Size | L2 Miss Rate | L2 Hit Time |
|---|---|---|---|
| a. | 1 MB | 95% | 5.62 ns |
| b. | 8 MB | 68% | 23.52 ns |

**5.7.4** [10] <5.3> What is the AMAT for P1 with the addition of an L2 cache? Is the AMAT better or worse with the L2 cache?

**5.7.5** [5] <5.3> Assuming a base CPI of 1.0 without any memory stalls, what is the total CPI for P1 with the addition of an L2 cache?

**5.7.6** [10] <5.3> Which processor is faster, now that P1 has an L2 cache? If P1 is faster, what miss rate would P2 need in its L1 cache to match P1's performance? If P2 is faster, what miss rate would P1 need in its L1 cache to match P2's performance?

5.7.6

a. P1 (4.15) is slower than P2 (2.68)

Let r be the miss rate:

AMAT: $1\times0.66 + 0.08\times5.62 + 0.08\times r\times70$ (ns)

$$AMAT(P1) * 0.36 + 1 * 0.64 \leq 2.68$$

Solve the equation.

# Exercise 5.8.1-5.8.3

## Exercise 5.8

This exercise examines the impact of different cache designs, specifically comparing associative caches to the direct-mapped caches from Section 5.2. For these exercises, refer to the table of address streams shown in Exercise 5.3.

**5.8.1** [10] <5.3> Using the references from Exercise 5.3, show the final cache contents for a three-way set associative cache with two-word blocks and a total size of 24 words. Use LRU replacement. For each reference identify the index bits, the tag bits, the block offset bits, and if it is a hit or a miss.

**5.8.2** [10] <5.3> Using the references from Exercise 5.3, show the final cache contents for a fully associative cache with one-word blocks and a total size of 8 words. Use LRU replacement. For each reference identify the index bits, the tag bits, and if it is a hit or a miss.

**5.8.3** [15] <5.3> Using the references from Exercise 5.3, what is the miss rate for a fully associative cache with two-word blocks and a total size of 8 words, using LRU replacement? What is the miss rate using MRU (most recently used) replacement? Finally what is the best possible miss rate for this cache, given any replacement policy?

| | Binary address | tag | index | h/m |
|---|---|---|---|---|
| 3 | 11 | 0 | 01 | miss |
| 180 | 10110100 | 10110 | 10 | miss |
| 43 | 101011 | 101 | 01 | miss |
| 2 | 10 | 0 | 01 | hit |
| 191 | 10111111 | 10111 | 11 | miss |
| 88 | 1011000 | 1011 | 00 | miss |
| 190 | 10111110 | 10111 | 11 | hit |
| 14 | 1110 | 1 | 11 | miss |
| 181 | 10110101 | 10110 | 10 | hit |
| 44 | 101100 | 101 | 10 | miss |
| 186 | 10111010 | 10111 | 01 | miss |
| 253 | 11111101 | 11111 | 10 | miss |

| Index | | | | | | |
|---|---|---|---|---|---|---|
| 00 | 88 | 89 | | | | |
| 01 | 2 | 3 | 42 | 43 | 186 | 187 |
| 10 | 180 | 181 | 44 | 45 | 252 | 253 |
| 11 | 190 | 191 | 14 | 15 | | |

bits 7–3 tag, 2–1 index, 0 block offset
The block offset is the address of the desired data within the block.

| | Binary address | tag | index | h/m |
|---|---|---|---|---|
| 3 | 11 | 11 | N/A | miss |
| 180 | 10110100 | 10110100 | N/A | miss |
| 43 | 101011 | 101011 | N/A | miss |
| 2 | 10 | 10 | N/A | miss |
| 191 | 10111111 | 10111111 | N/A | miss |
| 88 | 1011000 | 1011000 | N/A | miss |
| 190 | 10111110 | 10111110 | N/A | miss |
| 14 | 1110 | 1110 | N/A | miss |
| 181 | 10110101 | 10110101 | N/A | miss |
| 44 | 101100 | 101100 | N/A | miss |
| 186 | 10111010 | 10111010 | N/A | miss |
| 253 | 11111101 | 11111101 | N/A | miss |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 3→181 | 180→44 | 43→186 | 2→253 | 191 | 88 | 190 | 14 |

LRU

| | Binary address | tag | index | h/m |
|---|---|---|---|---|
| 3 | 11 | 11 | N/A | miss |
| 180 | 10110100 | 10110100 | N/A | miss |
| 43 | 101011 | 101011 | N/A | miss |
| 2 | 10 | 10 | N/A | hit |
| 191 | 10111111 | 10111111 | N/A | miss |
| 88 | 1011000 | 1011000 | N/A | miss |
| 190 | 10111110 | 10111110 | N/A | hit |
| 14 | 1110 | 1110 | N/A | miss |
| 181 | 10110101 | 10110101 | N/A | miss |
| 44 | 101100 | 101100 | N/A | miss |
| 186 | 10111010 | 10111010 | N/A | miss |
| 253 | 11111101 | 11111101 | N/A | miss |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 2→14→186 | 3→15→187 | 180→88→180→252 | 181→89→181→253 | 42→14 | 43→15 | 190→44 | 191→45 |

MRU

| | Binary address | tag | index | h/m |
|---|---|---|---|---|
| 3 | 11 | 11 | N/A | miss |
| 180 | 10110100 | 10110100 | N/A | miss |
| 43 | 101011 | 101011 | N/A | miss |
| 2 | 10 | 10 | N/A | hit |
| 191 | 10111111 | 10111111 | N/A | miss |
| 88 | 1011000 | 1011000 | N/A | miss |
| 190 | 10111110 | 10111110 | N/A | miss |
| 14 | 1110 | 1110 | N/A | miss |
| 181 | 10110101 | 10110101 | N/A | hit |
| 44 | 101100 | 101100 | N/A | miss |
| 186 | 10111010 | 10111010 | N/A | miss |
| 253 | 11111101 | 11111101 | N/A | miss |

The best miss rate is $\frac{10}{12}$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 2 | 3 | 180→44→186→253 | 181→45→187→253 | 42 | 43 | 190→88→190→14 | 191→89→191→15 |

# Exercise 5.8.4

Multilevel caching is an important technique to overcome the limited amount of space that a first level cache can provide while still maintaining its speed. Consider a processor with the following parameters:

| | Base CPI, No Memory Stalls | Processor Speed | Main Memory Access Time | First Level Cache Miss Rate per Instruction | Second Level Cache, Direct-Mapped Speed | Global Miss Rate with Second Level Cache, Direct-Mapped | Second Level Cache, Eight-Way Set Associative Speed | Global Miss Rate with Second Level Cache, Eight-Way Set Associative |
|---|---|---|---|---|---|---|---|---|
| **a.** | 1.5 | 2 GHz | 100 ns | 7% | 12 cycles | 3.5% | 28 cycles | 1.5% |
| **b.** | 1.0 | 2 GHz | 150 ns | 3% | 15 cycles | 5.0% | 20 cycles | 2.0% |

**5.8.4** [10] <5.3> Calculate the CPI for the processor in the table using: 1) only a first level cache, 2) a second level direct-mapped cache, and 3) a second level eight-way set associative cache. How do these numbers change if main memory access time is doubled? If it is cut in half?

1. Total CPI = base CPI + memory miss cycles × 1st level cache miss rate
2. Total CPI = base CPI + memory miss cycles × global miss rate w/2nd level direct-mapped cache + 2nd level direct-mapped speed × 1st level cache miss rate
3. Total CPI = base CPI + memory miss cycles × global miss rate w/2nd level 8-way set assoc cache + 2nd level 8-way set assoc speed × 1st level cache miss rate

Base CPI: 1.5

Memory miss cycles: $100\ ns \times 3GHz = 300\ clock\ cycles$

1. Total CPI: $1.5 + 300 \times 7\% = 22.5/43.5/12\ (normal/double/half)$
2. Total CPI: $1.5 + 12 \times 7\% + 300 \times 3.5\% = 12.84/23.34/7.59$
3. Total CPI: $1.5 + 28 \times 7\% + 300 \times 1.5\% = 13.96/24.46/8.71$