

```
for (i = 0; i < 4; i++)
{
    for (j = 0; j < 4; j++)
    {
        acc = 0;
        for (k = 0, k < 4; k++)
        {
            acc = acc + A[i][k] * B[k][j];
        }
        C[i][j] = acc;
    }
}
```

@A=400  
@B=440  
@C=480

Tag		Index		Offset	
31	8	7	4	3	0

V	Tag	Index				
0		F				
0		E				
0		D				
0		C				
0		B				
0		A				
0		9				
0		8				
0		7				
0		6				
0		5				
0		4				
0		3				
0		2				
0		1				
0		0				

C[3,3] @4BC	C[3,2] @4B8	C[3,1] @4B4	C[3,0] @4B0
C[2,3] @4AC	C[2,2] @4A8	C[2,1] @4A4	C[2,0] @4A0
C[1,3] @49C	C[1,2] @498	C[1,1] @494	C[1,0] @490
C[0,3] @48C	C[0,2] @488	C[0,1] @484	C[0,0] @480
B[3,3] @47C	B[3,2] @478	B[3,1] @474	B[3,0] @470
B[2,3] @46C	B[2,2] @468	B[2,1] @464	B[2,0] @460
B[1,3] @45C	B[1,2] @458	B[1,1] @454	B[1,0] @450
B[0,3] @44C	B[0,2] @448	B[0,1] @444	B[0,0] @440
A[3,3] @43C	A[3,2] @438	A[3,1] @434	A[3,0] @430
A[2,3] @42C	A[2,2] @428	A[2,1] @424	A[2,0] @420
A[1,3] @41C	A[1,2] @418	A[1,1] @414	A[1,0] @410
A[0,3] @40C	A[0,2] @408	A[0,1] @404	A[0,0] @400

```
for (i = 0; i < 4; i++)      i=0
{
    for (j = 0; j < 4; j++)  j=0
    {
        acc = 0;
        for (k = 0; k < 4; k++) k=0
        {
            acc = acc + A[i][k] * B[k][j];  Acc=acc+A[00]*B[00]
        }
        C[i][j] = acc;
    }
}
```

← Accès à l'adresse 400

Tag		Index		Offset	
31	8	7	4	3	0

V	Tag	Index				
0		F				
0		E				
0		D				
0		C				
0		B				
0		A				
0		9				
0		8				
0		7				
0		6				
0		5				
0		4				
0		3				
0		2				
0		1				
0		0				

C[3,3] @4BC	C[3,2] @4B8	C[3,1] @4B4	C[3,0] @4B0
C[2,3] @4AC	C[2,2] @4A8	C[2,1] @4A4	C[2,0] @4A0
C[1,3] @49C	C[1,2] @498	C[1,1] @494	C[1,0] @490
C[0,3] @48C	C[0,2] @488	C[0,1] @484	C[0,0] @480
B[3,3] @47C	B[3,2] @478	B[3,1] @474	B[3,0] @470
B[2,3] @46C	B[2,2] @468	B[2,1] @464	B[2,0] @460
B[1,3] @45C	B[1,2] @458	B[1,1] @454	B[1,0] @450
B[0,3] @44C	B[0,2] @448	B[0,1] @444	B[0,0] @440
A[3,3] @43C	A[3,2] @438	A[3,1] @434	A[3,0] @430
A[2,3] @42C	A[2,2] @428	A[2,1] @424	A[2,0] @420
A[1,3] @41C	A[1,2] @418	A[1,1] @414	A[1,0] @410
A[0,3] @40C	A[0,2] @408	A[0,1] @404	A[0,0] @400

```

for (i = 0; i < 4; i++)      i=0
{
    for (j = 0; j < 4; j++)  j=0
    {
        acc = 0;
        for (k = 0; k < 4; k++) k=0
        {
            acc = acc + A[i][k] * B[k][j];  Acc=acc+A[00]*B[00]
        }
        C[i][j] = acc;
    }
}

```

← Accès à l'adresse 400

Tag	Index	Offset
31 4 8	7 0 4	3 0 0

V	Tag	Index				
0		F				
0		E				
0		D				
0		C				
0		B				
0		A				
0		9				
0		8				
0		7				
0		6				
0		5				
0		4				
0		3				
0		2				
0		1				
0		0				

- À l'index 0 dans la mémoire cache, le bit de validité est à 0, c'est un échec.

C[3,3] @4BC	C[3,2] @4B8	C[3,1] @4B4	C[3,0] @4B0
C[2,3] @4AC	C[2,2] @4A8	C[2,1] @4A4	C[2,0] @4A0
C[1,3] @49C	C[1,2] @498	C[1,1] @494	C[1,0] @490
C[0,3] @48C	C[0,2] @488	C[0,1] @484	C[0,0] @480
B[3,3] @47C	B[3,2] @478	B[3,1] @474	B[3,0] @470
B[2,3] @46C	B[2,2] @468	B[2,1] @464	B[2,0] @460
B[1,3] @45C	B[1,2] @458	B[1,1] @454	B[1,0] @450
B[0,3] @44C	B[0,2] @448	B[0,1] @444	B[0,0] @440
A[3,3] @43C	A[3,2] @438	A[3,1] @434	A[3,0] @430
A[2,3] @42C	A[2,2] @428	A[2,1] @424	A[2,0] @420
A[1,3] @41C	A[1,2] @418	A[1,1] @414	A[1,0] @410
A[0,3] @40C	A[0,2] @408	A[0,1] @404	A[0,0] @400

```

for (i = 0; i < 4; i++)      i=0
{
  for (j = 0; j < 4; j++)    j=0
  {
    acc = 0;
    for (k = 0; k < 4; k++)  k=0
    {
      acc = acc + A[i][k] * B[k][j];  Acc=acc+A[00]*B[00]
    }
    C[i][j] = acc;
  }
}

```

← Accès à l'adresse 400

Tag	Index	Offset
31 4 8	7 0 4	3 0 0

V	Tag	Index				
0		F				
0		E				
0		D				
0		C				
0		B				
0		A				
0		9				
0		8				
0		7				
0		6				
0		5				
0		4				
0		3				
0		2				
0		1				
0		0				

- À l'index 0 dans la mémoire cache, le bit de validité est à 0, c'est un échec.
- Le blco de 16 octets de la mémoire centrale correspondant à l'adresse 400 est rapatrié en mém. cache

C[3,3] @4BC	C[3,2] @4B8	C[3,1] @4B4	C[3,0] @4B0
C[2,3] @4AC	C[2,2] @4A8	C[2,1] @4A4	C[2,0] @4A0
C[1,3] @49C	C[1,2] @498	C[1,1] @494	C[1,0] @490
C[0,3] @48C	C[0,2] @488	C[0,1] @484	C[0,0] @480
B[3,3] @47C	B[3,2] @478	B[3,1] @474	B[3,0] @470
B[2,3] @46C	B[2,2] @468	B[2,1] @464	B[2,0] @460
B[1,3] @45C	B[1,2] @458	B[1,1] @454	B[1,0] @450
B[0,3] @44C	B[0,2] @448	B[0,1] @444	B[0,0] @440
A[3,3] @43C	A[3,2] @438	A[3,1] @434	A[3,0] @430
A[2,3] @42C	A[2,2] @428	A[2,1] @424	A[2,0] @420
A[1,3] @41C	A[1,2] @418	A[1,1] @414	A[1,0] @410
A[0,3] @40C	A[0,2] @408	A[0,1] @404	A[0,0] @400

```

for (i = 0; i < 4; i++)      i=0
{
    for (j = 0; j < 4; j++)  j=0
    {
        acc = 0;
        for (k = 0; k < 4; k++) k=0
        {
            acc = acc + A[i][k] * B[k][j];  Acc=acc+A[00]*B[00]
        }
        C[i][j] = acc;
    }
}

```

← Accès à l'adresse 400

Tag	Index	Offset
31 4 8	7 0 4	3 0 0

V	Tag	Index				
0		F				
0		E				
0		D				
0		C				
0		B				
0		A				
0		9				
0		8				
0		7				
0		6				
0		5				
0		4				
0		3				
0		2				
0		1				
1	4	0	A[0,3]	A[0,2]	A[0,1]	A[0,0]

- À l'index 0 dans la mémoire cache, le bit de validité est à 0, c'est un échec.
- Le blco de 16 octets de la mémoire centrale correspondant à l'adresse 400 est rapatrié en mém. Cache
- La mémoire cache est mise à jour pour l'index 0

C[3,3] @4BC	C[3,2] @4B8	C[3,1] @4B4	C[3,0] @4B0
C[2,3] @4AC	C[2,2] @4A8	C[2,1] @4A4	C[2,0] @4A0
C[1,3] @49C	C[1,2] @498	C[1,1] @494	C[1,0] @490
C[0,3] @48C	C[0,2] @488	C[0,1] @484	C[0,0] @480
B[3,3] @47C	B[3,2] @478	B[3,1] @474	B[3,0] @470
B[2,3] @46C	B[2,2] @468	B[2,1] @464	B[2,0] @460
B[1,3] @45C	B[1,2] @458	B[1,1] @454	B[1,0] @450
B[0,3] @44C	B[0,2] @448	B[0,1] @444	B[0,0] @440
A[3,3] @43C	A[3,2] @438	A[3,1] @434	A[3,0] @430
A[2,3] @42C	A[2,2] @428	A[2,1] @424	A[2,0] @420
A[1,3] @41C	A[1,2] @418	A[1,1] @414	A[1,0] @410
A[0,3] @40C	A[0,2] @408	A[0,1] @404	A[0,0] @400

```

for (i = 0; i < 4; i++)      i=0
{
    for (j = 0; j < 4; j++)  j=0
    {
        acc = 0;
        for (k = 0; k < 4; k++) k=0
        {
            acc = acc + A[i][k] * B[k][j];  Acc=acc+A[00]*B[00]
        }
        C[i][j] = acc;
    }
}

```

← Accès à l'adresse 440

Tag	Index	Offset
31 4 8	7 4 4	3 0 0

V	Tag	Index				
0		F				
0		E				
0		D				
0		C				
0		B				
0		A				
0		9				
0		8				
0		7				
0		6				
0		5				
0		4				
0		3				
0		2				
0		1				
1	4	0	A[0,3]	A[0,2]	A[0,1]	A[0,0]

- À l'index 0 dans la mémoire cache, le bit de validité est à 0, c'est un échec.
- Le bloc de 16 octets de la mémoire centrale correspondant à l'adresse 440 est rapatrié en mém. Cache
- La mémoire cache est mise à jour pour l'index 0

C[3,3] @4BC	C[3,2] @4B8	C[3,1] @4B4	C[3,0] @4B0
C[2,3] @4AC	C[2,2] @4A8	C[2,1] @4A4	C[2,0] @4A0
C[1,3] @49C	C[1,2] @498	C[1,1] @494	C[1,0] @490
C[0,3] @48C	C[0,2] @488	C[0,1] @484	C[0,0] @480
B[3,3] @47C	B[3,2] @478	B[3,1] @474	B[3,0] @470
B[2,3] @46C	B[2,2] @468	B[2,1] @464	B[2,0] @460
B[1,3] @45C	B[1,2] @458	B[1,1] @454	B[1,0] @450
B[0,3] @44C	B[0,2] @448	B[0,1] @444	B[0,0] @440
A[3,3] @43C	A[3,2] @438	A[3,1] @434	A[3,0] @430
A[2,3] @42C	A[2,2] @428	A[2,1] @424	A[2,0] @420
A[1,3] @41C	A[1,2] @418	A[1,1] @414	A[1,0] @410
A[0,3] @40C	A[0,2] @408	A[0,1] @404	A[0,0] @400

```

for (i = 0; i < 4; i++)      i=0
{
    for (j = 0; j < 4; j++)  j=0
    {
        acc = 0;
        for (k = 0; k < 4; k++) k=0
        {
            acc = acc + A[i][k] * B[k][j];  Acc=acc+A[00]*B[00]
        }
        C[i][j] = acc;
    }
}

```

← Accès à l'adresse 440

Tag	Index	Offset
31 4 8	7 4 4	3 0 0

V	Tag	Index				
0		F				
0		E				
0		D				
0		C				
0		B				
0		A				
0		9				
0		8				
0		7				
0		6				
0		5				
0		4				
0		3				
0		2				
0		1				
1	4	0	A[0,3]	A[0,2]	A[0,1]	A[0,0]

- À l'index 0 dans la mémoire cache, le bit de validité est à 0, c'est un échec.
- Le bloc de 16 octets de la mémoire centrale correspondant à l'adresse 440 est rapatrié en mém. Cache

C[3,3] @4BC	C[3,2] @4B8	C[3,1] @4B4	C[3,0] @4B0
C[2,3] @4AC	C[2,2] @4A8	C[2,1] @4A4	C[2,0] @4A0
C[1,3] @49C	C[1,2] @498	C[1,1] @494	C[1,0] @490
C[0,3] @48C	C[0,2] @488	C[0,1] @484	C[0,0] @480
B[3,3] @47C	B[3,2] @478	B[3,1] @474	B[3,0] @470
B[2,3] @46C	B[2,2] @468	B[2,1] @464	B[2,0] @460
B[1,3] @45C	B[1,2] @458	B[1,1] @454	B[1,0] @450
B[0,3] @44C	B[0,2] @448	B[0,1] @444	B[0,0] @440
A[3,3] @43C	A[3,2] @438	A[3,1] @434	A[3,0] @430
A[2,3] @42C	A[2,2] @428	A[2,1] @424	A[2,0] @420
A[1,3] @41C	A[1,2] @418	A[1,1] @414	A[1,0] @410
A[0,3] @40C	A[0,2] @408	A[0,1] @404	A[0,0] @400

```

for (i = 0; i < 4; i++)      i=0
{
    for (j = 0; j < 4; j++)  j=0
    {
        acc = 0;
        for (k = 0; k < 4; k++) k=0
        {
            acc = acc + A[i][k] * B[k][j];  Acc=acc+A[00]*B[00]
        }
        C[i][j] = acc;
    }
}

```

← Accès à l'adresse 440

Tag	Index	Offset
31 4 8	7 4 4	3 0 0

V	Tag	Index				
0		F				
0		E				
0		D				
0		C				
0		B				
0		A				
0		9				
0		8				
0		7				
0		6				
0		5				
1	4	4	B[0,3]	B[0,2]	B[0,1]	B[0,0]
0		3				
0		2				
0		1				
1	4	0	A[0,3]	A[0,2]	A[0,1]	A[0,0]

- À l'index 0 dans la mémoire cache, le bit de validité est à 0, c'est un échec.
- Le bloc de 16 octets de la mémoire centrale correspondant à l'adresse 440 est rapatrié en mém. Cache
- La mémoire cache est mise à jour pour l'index 4

C[3,3] @4BC	C[3,2] @4B8	C[3,1] @4B4	C[3,0] @4B0
C[2,3] @4AC	C[2,2] @4A8	C[2,1] @4A4	C[2,0] @4A0
C[1,3] @49C	C[1,2] @498	C[1,1] @494	C[1,0] @490
C[0,3] @48C	C[0,2] @488	C[0,1] @484	C[0,0] @480
B[3,3] @47C	B[3,2] @478	B[3,1] @474	B[3,0] @470
B[2,3] @46C	B[2,2] @468	B[2,1] @464	B[2,0] @460
B[1,3] @45C	B[1,2] @458	B[1,1] @454	B[1,0] @450
B[0,3] @44C	B[0,2] @448	B[0,1] @444	B[0,0] @440
A[3,3] @43C	A[3,2] @438	A[3,1] @434	A[3,0] @430
A[2,3] @42C	A[2,2] @428	A[2,1] @424	A[2,0] @420
A[1,3] @41C	A[1,2] @418	A[1,1] @414	A[1,0] @410
A[0,3] @40C	A[0,2] @408	A[0,1] @404	A[0,0] @400



```

for (i = 0; i < 4; i++)      i=0
{
    for (j = 0; j < 4; j++)  j=0
    {
        acc = 0;
        for (k = 0; k < 4; k++) k=0
        {
            acc = acc + A[i][k] * B[k][j];  Acc=acc+A[00]*B[00]
        }
        C[i][j] = acc;
    }
}

```

C[00]=acc  
↙  
Accès à l'adresse 480

Tag	Index	Offset
31 <b>4</b> 8	7 <b>8</b> 4	3 <b>0</b> 0

V	Tag	Index				
0		F				
0		E				
0		D				
0		C				
0		B				
0		A				
0		9				
0		8				
0		7				
0		6				
0		5				
1	4	4	B[0,3]	B[0,2]	B[0,1]	B[0,0]
0		3				
0		2				
0		1				
1	4	0	A[0,3]	A[0,2]	A[0,1]	A[0,0]

- À l’index 8 dans la mémoire cache, le bit de validité est à 0, c’est un échec.
- Le bloc de 16 octets de la mémoire centrale correspondant à l’adresse 480 est rapatrié en mém. Cache
- La mémoire cache est mise à jour pour l’index 8

C[3,3] @4BC	C[3,2] @4B8	C[3,1] @4B4	C[3,0] @4B0
C[2,3] @4AC	C[2,2] @4A8	C[2,1] @4A4	C[2,0] @4A0
C[1,3] @49C	C[1,2] @498	C[1,1] @494	C[1,0] @490
<span style="color: red;">C[0,3] @48C</span>	<span style="color: red;">C[0,2] @488</span>	<span style="color: red;">C[0,1] @484</span>	<span style="color: red;">C[0,0] @480</span>
B[3,3] @47C	B[3,2] @478	B[3,1] @474	B[3,0] @470
B[2,3] @46C	B[2,2] @468	B[2,1] @464	B[2,0] @460
B[1,3] @45C	B[1,2] @458	B[1,1] @454	B[1,0] @450
B[0,3] @44C	B[0,2] @448	B[0,1] @444	B[0,0] @440
A[3,3] @43C	A[3,2] @438	A[3,1] @434	A[3,0] @430
A[2,3] @42C	A[2,2] @428	A[2,1] @424	A[2,0] @420
A[1,3] @41C	A[1,2] @418	A[1,1] @414	A[1,0] @410
A[0,3] @40C	A[0,2] @408	A[0,1] @404	A[0,0] @400

```

for (i = 0; i < 4; i++)      i=0
{
    for (j = 0; j < 4; j++)  j=0
    {
        acc = 0;
        for (k = 0; k < 4; k++) k=0
        {
            acc = acc + A[i][k] * B[k][j];  Acc=acc+A[00]*B[00]
        }
        C[i][j] = acc;
    }
}

```

C[00]=acc

Accès à l'adresse 480

Tag	Index	Offset
31 4 8	7 8 4	3 0 0

V	Tag	Index				
0		F				
0		E				
0		D				
0		C				
0		B				
0		A				
0		9				
0	4	8	C[0,3]	C[0,2]	C[0,1]	C[0,0]
0		7				
0		6				
0		5				
1	4	4	B[0,3]	B[0,2]	B[0,1]	B[0,0]
0		3				
0		2				
0		1				
1	4	0	A[0,3]	A[0,2]	A[0,1]	A[0,0]

- À l'index 8 dans la mémoire cache, le bit de validité est à 0, c'est un échec.
- Le bloc de 16 octets de la mémoire centrale correspondant à l'adresse 480 est rapatrié en mém. Cache
- La mémoire cache est mise à jour pour l'index 8

C[3,3] @4BC	C[3,2] @4B8	C[3,1] @4B4	C[3,0] @4B0
C[2,3] @4AC	C[2,2] @4A8	C[2,1] @4A4	C[2,0] @4A0
C[1,3] @49C	C[1,2] @498	C[1,1] @494	C[1,0] @490
C[0,3] @48C	C[0,2] @488	C[0,1] @484	C[0,0] @480
B[3,3] @47C	B[3,2] @478	B[3,1] @474	B[3,0] @470
B[2,3] @46C	B[2,2] @468	B[2,1] @464	B[2,0] @460
B[1,3] @45C	B[1,2] @458	B[1,1] @454	B[1,0] @450
B[0,3] @44C	B[0,2] @448	B[0,1] @444	B[0,0] @440
A[3,3] @43C	A[3,2] @438	A[3,1] @434	A[3,0] @430
A[2,3] @42C	A[2,2] @428	A[2,1] @424	A[2,0] @420
A[1,3] @41C	A[1,2] @418	A[1,1] @414	A[1,0] @410
A[0,3] @40C	A[0,2] @408	A[0,1] @404	A[0,0] @400

```

for (i = 0; i < 4; i++)      i=0
{
    for (j = 0; j < 4; j++)  j=0
    {
        acc = 0;
        for (k = 0; k < 4; k++) k=1
        {
            acc = acc + A[i][k] * B[k][j];  Acc=acc+A[01]*B[10]
        }
        C[i][j] = acc;
    }
}

```

← Accès à l'adresse 404

Tag	Index	Offset
31 4 8	7 0 4	3 4 0

V	Tag	Index				
0		F				
0		E				
0		D				
0		C				
0		B				
0		A				
0		9				
0	4	8	C[0,3]	C[0,2]	C[0,1]	C[0,0]
0		7				
0		6				
0		5				
1	4	4	B[0,3]	B[0,2]	B[0,1]	B[0,0]
0		3				
0		2				
0		1				
1	4	0	A[0,3]	A[0,2]	A[0,1]	A[0,0]

A l’index 0, en mémoire cache, le bit de validité vaut 1 et le tag vaut 4

C[3,3] @4BC	C[3,2] @4B8	C[3,1] @4B4	C[3,0] @4B0
C[2,3] @4AC	C[2,2] @4A8	C[2,1] @4A4	C[2,0] @4A0
C[1,3] @49C	C[1,2] @498	C[1,1] @494	C[1,0] @490
C[0,3] @48C	C[0,2] @488	C[0,1] @484	C[0,0] @480
B[3,3] @47C	B[3,2] @478	B[3,1] @474	B[3,0] @470
B[2,3] @46C	B[2,2] @468	B[2,1] @464	B[2,0] @460
B[1,3] @45C	B[1,2] @458	B[1,1] @454	B[1,0] @450
B[0,3] @44C	B[0,2] @448	B[0,1] @444	B[0,0] @440
A[3,3] @43C	A[3,2] @438	A[3,1] @434	A[3,0] @430
A[2,3] @42C	A[2,2] @428	A[2,1] @424	A[2,0] @420
A[1,3] @41C	A[1,2] @418	A[1,1] @414	A[1,0] @410
A[0,3] @40C	A[0,2] @408	A[0,1] @404	A[0,0] @400

```

for (i = 0; i < 4; i++)      i=0
{
    for (j = 0; j < 4; j++)  j=0
    {
        acc = 0;
        for (k = 0; k < 4; k++) k=1
        {
            acc = acc + A[i][k] * B[k][j];  Acc=acc+A[01]*B[10]
        }
        C[i][j] = acc;
    }
}

```

← Accès à l'adresse 404

Tag	Index	Offset
31 4 8	7 0 4	3 4 0

V	Tag	Index				
0		F				
0		E				
0		D				
0		C				
0		B				
0		A				
0		9				
0	4	8	C[0,3]	C[0,2]	C[0,1]	C[0,0]
0		7				
0		6				
0		5				
1	4	4	B[0,3]	B[0,2]	B[0,1]	B[0,0]
0		3				
0		2				
0		1				
1	4	0	A[0,3]	A[0,2]	A[0,1]	A[0,0]

- A l'index 0, en mémoire cache, le bit de validité vaut 1 et le tag vaut 4, la donnée que l'on cherche se trouve dans le cache à l'index 0. C'est un succès.

C[3,3] @4BC	C[3,2] @4B8	C[3,1] @4B4	C[3,0] @4B0
C[2,3] @4AC	C[2,2] @4A8	C[2,1] @4A4	C[2,0] @4A0
C[1,3] @49C	C[1,2] @498	C[1,1] @494	C[1,0] @490
C[0,3] @48C	C[0,2] @488	C[0,1] @484	C[0,0] @480
B[3,3] @47C	B[3,2] @478	B[3,1] @474	B[3,0] @470
B[2,3] @46C	B[2,2] @468	B[2,1] @464	B[2,0] @460
B[1,3] @45C	B[1,2] @458	B[1,1] @454	B[1,0] @450
B[0,3] @44C	B[0,2] @448	B[0,1] @444	B[0,0] @440
A[3,3] @43C	A[3,2] @438	A[3,1] @434	A[3,0] @430
A[2,3] @42C	A[2,2] @428	A[2,1] @424	A[2,0] @420
A[1,3] @41C	A[1,2] @418	A[1,1] @414	A[1,0] @410
A[0,3] @40C	A[0,2] @408	A[0,1] @404	A[0,0] @400

```

for (i = 0; i < 4; i++)      i=0
{
    for (j = 0; j < 4; j++)  j=0
    {
        acc = 0;
        for (k = 0; k < 4; k++) k=1
        {
            acc = acc + A[i][k] * B[k][j];  Acc=acc+A[01]*B[10]
        }
        C[i][j] = acc;
    }
}

```

← Accès à l'adresse 404

Tag			Index			Offset		
31	4	8	7	0	4	3	4	0

V	Tag	Index				
0		F				
0		E				
0		D				
0		C				
0		B				
0		A				
0		9				
0	4	8	C[0,3]	C[0,2]	C[0,1]	C[0,0]
0		7				
0		6				
0		5				
1	4	4	B[0,3]	B[0,2]	B[0,1]	B[0,0]
0		3				
0		2				
0		1				
1	4	0	A[0,3]	A[0,2]	A[0,1]	A[0,0]

- A l'index 0, en mémoire cache, le bit de validité vaut 1 et le tag vaut 4, la donnée que l'on cherche se trouve dans le cache à l'index 0. C'est un succès.
- La valeur de l'offset, nous indique que la donnée recherché par le processeur est le deuxieme mot.

C[3,3] @4BC	C[3,2] @4B8	C[3,1] @4B4	C[3,0] @4B0
C[2,3] @4AC	C[2,2] @4A8	C[2,1] @4A4	C[2,0] @4A0
C[1,3] @49C	C[1,2] @498	C[1,1] @494	C[1,0] @490
C[0,3] @48C	C[0,2] @488	C[0,1] @484	C[0,0] @480
B[3,3] @47C	B[3,2] @478	B[3,1] @474	B[3,0] @470
B[2,3] @46C	B[2,2] @468	B[2,1] @464	B[2,0] @460
B[1,3] @45C	B[1,2] @458	B[1,1] @454	B[1,0] @450
B[0,3] @44C	B[0,2] @448	B[0,1] @444	B[0,0] @440
A[3,3] @43C	A[3,2] @438	A[3,1] @434	A[3,0] @430
A[2,3] @42C	A[2,2] @428	A[2,1] @424	A[2,0] @420
A[1,3] @41C	A[1,2] @418	A[1,1] @414	A[1,0] @410
A[0,3] @40C	A[0,2] @408	A[0,1] @404	A[0,0] @400

```
for (i = 0; i < 4; i++)
{
    for (j = 0; j < 4; j++)
    {
        acc = 0;
        for (k = 0; k < 4; k++)
        {
            acc = acc + A[i][k] * B[k][j];
        }
        C[i][j] = acc;
    }
}
```

@A=400  
@B=800  
@C=480

Tag		Index		Offset	
31	8	7	4	3	0

V	Tag	Index				
0		F				
0		E				
0		D				
0		C				
0		B				
0		A				
0		9				
0		8				
0		7				
0		6				
0		5				
0		4				
0		3				
0		2				
0		1				
0		0				

B[3,3] @83C	B[3,2] @838	B[3,1] @834	B[3,0] @830
B[2,3] @82C	B[2,2] @828	B[2,1] @824	B[2,0] @820
B[1,3] @81C	B[1,2] @818	B[1,1] @814	B[1,0] @810
B[0,3] @80C	B[0,2] @808	B[0,1] @804	B[0,0] @800
...	...	...	....
C[3,3] @4BC	C[3,2] @4B8	C[3,1] @4B4	C[3,0] @4B0
C[2,3] @4AC	C[2,2] @4A8	C[2,1] @4A4	C[2,0] @4A0
C[1,3] @49C	C[1,2] @498	C[1,1] @494	C[1,0] @490
C[0,3] @48C	C[0,2] @488	C[0,1] @484	C[0,0] @480
...	...	...	....
A[3,3] @43C	A[3,2] @438	A[3,1] @434	A[3,0] @430
A[2,3] @42C	A[2,2] @428	A[2,1] @424	A[2,0] @420
A[1,3] @41C	A[1,2] @418	A[1,1] @414	A[1,0] @410
A[0,3] @40C	A[0,2] @408	A[0,1] @404	A[0,0] @400

```

for (i = 0; i < 4; i++)      i=0
{
    for (j = 0; j < 4; j++)  j=0
    {
        acc = 0;
        for (k = 0; k < 4; k++) k=0
        {
            acc = acc + A[i][k] * B[k][j];  Acc=acc+A[00]*B[00]
        }
        C[i][j] = acc;
    }
}

```

← Accès à l'adresse 400

Tag		Index		Offset	
31	8	7	4	3	0

V	Tag	Index				
0		F				
0		E				
0		D				
0		C				
0		B				
0		A				
0		9				
0		8				
0		7				
0		6				
0		5				
0		4				
0		3				
0		2				
0		1				
0		0				

B[3,3] @83C	B[3,2] @838	B[3,1] @834	B[3,0] @830
B[2,3] @82C	B[2,2] @828	B[2,1] @824	B[2,0] @820
B[1,3] @81C	B[1,2] @818	B[1,1] @814	B[1,0] @810
B[0,3] @80C	B[0,2] @808	B[0,1] @804	B[0,0] @800
...	...	...	....
C[3,3] @4BC	C[3,2] @4B8	C[3,1] @4B4	C[3,0] @4B0
C[2,3] @4AC	C[2,2] @4A8	C[2,1] @4A4	C[2,0] @4A0
C[1,3] @49C	C[1,2] @498	C[1,1] @494	C[1,0] @490
C[0,3] @48C	C[0,2] @488	C[0,1] @484	C[0,0] @480
...	...	...	....
A[3,3] @43C	A[3,2] @438	A[3,1] @434	A[3,0] @430
A[2,3] @42C	A[2,2] @428	A[2,1] @424	A[2,0] @420
A[1,3] @41C	A[1,2] @418	A[1,1] @414	A[1,0] @410
A[0,3] @40C	A[0,2] @408	A[0,1] @404	A[0,0] @400

```

for (i = 0; i < 4; i++)      i=0
{
    for (j = 0; j < 4; j++)  j=0
    {
        acc = 0;
        for (k = 0; k < 4; k++) k=0
        {
            acc = acc + A[i][k] * B[k][j];  Acc=acc+A[00]*B[00]
        }
        C[i][j] = acc;
    }
}

```

← Accès à l'adresse 400

Tag	Index	Offset
31 <b>4</b> 8	7 <b>0</b> 4	3 <b>0</b> 0

V	Tag	Index				
0		F				
0		E				
0		D				
0		C				
0		B				
0		A				
0		9				
0		8				
0		7				
0		6				
0		5				
0		4				
0		3				
0		2				
0		1				
<b>1</b>	<b>4</b>	<b>0</b>	A[0,3]	A[0,2]	A[0,1]	A[0,0]

- À l'index 0 dans la mémoire cache, le bit de validité est à 0, c'est un échec.
- Le bloc de 16 octets de la mémoire centrale correspondant à l'adresse 400 est rapatrié en mém. Cache
- La mémoire cache est mise à jour pour l'index 0

B[3,3] @83C	B[3,2] @838	B[3,1] @834	B[3,0] @830
B[2,3] @82C	B[2,2] @828	B[2,1] @824	B[2,0] @820
B[1,3] @81C	B[1,2] @818	B[1,1] @814	B[1,0] @810
B[0,3] @80C	B[0,2] @808	B[0,1] @804	B[0,0] @800
...	...	...	....
C[3,3] @4BC	C[3,2] @4B8	C[3,1] @4B4	C[3,0] @4B0
C[2,3] @4AC	C[2,2] @4A8	C[2,1] @4A4	C[2,0] @4A0
C[1,3] @49C	C[1,2] @498	C[1,1] @494	C[1,0] @490
C[0,3] @48C	C[0,2] @488	C[0,1] @484	C[0,0] @480
...	...	...	....
A[3,3] @43C	A[3,2] @438	A[3,1] @434	A[3,0] @430
A[2,3] @42C	A[2,2] @428	A[2,1] @424	A[2,0] @420
A[1,3] @41C	A[1,2] @418	A[1,1] @414	A[1,0] @410
A[0,3] @40C	A[0,2] @408	A[0,1] @404	A[0,0] @400



```

for (i = 0; i < 4; i++)      i=0
{
    for (j = 0; j < 4; j++)  j=0
    {
        acc = 0;
        for (k = 0; k < 4; k++) k=0
        {
            acc = acc + A[i][k] * B[k][j];  Acc=acc+A[00]*B[00]
        }
        C[i][j] = acc;
    }
}

```

← Accès à l'adresse 800

Tag	Index	Offset
31 8 8	7 0 4	3 0 0

V	Tag	Index				
0		F				
0		E				
0		D				
0		C				
0		B				
0		A				
0		9				
0		8				
0		7				
0		6				
0		5				
0		4				
0		3				
0		2				
0		1				
1	4	0	A[0,3]	A[0,2]	A[0,1]	A[0,0]

- À l'index 0 dans la mémoire cache, le bit de validité est à 1, Mais le tag vaut 4

B[3,3] @83C	B[3,2] @838	B[3,1] @834	B[3,0] @830
B[2,3] @82C	B[2,2] @828	B[2,1] @824	B[2,0] @820
B[1,3] @81C	B[1,2] @818	B[1,1] @814	B[1,0] @810
B[0,3] @80C	B[0,2] @808	B[0,1] @804	B[0,0] @800
...	...	...	....
C[3,3] @4BC	C[3,2] @4B8	C[3,1] @4B4	C[3,0] @4B0
C[2,3] @4AC	C[2,2] @4A8	C[2,1] @4A4	C[2,0] @4A0
C[1,3] @49C	C[1,2] @498	C[1,1] @494	C[1,0] @490
C[0,3] @48C	C[0,2] @488	C[0,1] @484	C[0,0] @480
...	...	...	....
A[3,3] @43C	A[3,2] @438	A[3,1] @434	A[3,0] @430
A[2,3] @42C	A[2,2] @428	A[2,1] @424	A[2,0] @420
A[1,3] @41C	A[1,2] @418	A[1,1] @414	A[1,0] @410
A[0,3] @40C	A[0,2] @408	A[0,1] @404	A[0,0] @400

```
for (i = 0; i < 4; i++)      i=0
{
  for (j = 0; j < 4; j++)    j=0
  {
    acc = 0;
    for (k = 0; k < 4; k++)  k=0
    {
      acc = acc + A[i][k] * B[k][j];  Acc=acc+A[00]*B[00]
    }
    C[i][j] = acc;
  }
}
```

← Accès à l'adresse 800

Tag	Index	Offset
21 8	7 0 4	3 0 0

V	Tag	Index				
0		F				
0		E				
0		D				
0		C				
0		B				
0		A				
0		9				
0		8				
0		7				
0		6				
0		5				
0		4				
0		3				
0		2				
0		1				
1	4	0	A[0,3]	A[0,2]	A[0,1]	A[0,0]

- À l'index 0 dans la mémoire cache, le bit de validité est à 1, Mais le tag vaut 4
- Il s'agit donc d'un échec car les données places dans le cache à l'index 0 ne sont pas celles correspondant à l'adresse 800

B[3,3] @83C	B[3,2] @838	B[3,1] @834	B[3,0] @830
B[2,3] @82C	B[2,2] @828	B[2,1] @824	B[2,0] @820
B[1,3] @81C	B[1,2] @818	B[1,1] @814	B[1,0] @810
B[0,3] @80C	B[0,2] @808	B[0,1] @804	B[0,0] @800
...	...	...	....
C[3,3] @4BC	C[3,2] @4B8	C[3,1] @4B4	C[3,0] @4B0
C[2,3] @4AC	C[2,2] @4A8	C[2,1] @4A4	C[2,0] @4A0
C[1,3] @49C	C[1,2] @498	C[1,1] @494	C[1,0] @490
C[0,3] @48C	C[0,2] @488	C[0,1] @484	C[0,0] @480
...	...	...	....
A[3,3] @43C	A[3,2] @438	A[3,1] @434	A[3,0] @430
A[2,3] @42C	A[2,2] @428	A[2,1] @424	A[2,0] @420
A[1,3] @41C	A[1,2] @418	A[1,1] @414	A[1,0] @410
A[0,3] @40C	A[0,2] @408	A[0,1] @404	A[0,0] @400

```
for (i = 0; i < 4; i++)      i=0
{
    for (j = 0; j < 4; j++)  j=0
    {
        acc = 0;
        for (k = 0; k < 4; k++) k=0
        {
            acc = acc + A[i][k] * B[k][j];  Acc=acc+A[00]*B[00]
        }
        C[i][j] = acc;
    }
}
```

← Accès à l'adresse 800

Tag		Index		Offset	
31	8	8	7	0	4
			3	0	0

V	Tag	Index				
0		F				
0		E				
0		D				
0		C				
0		B				
0		A				
0		9				
0		8				
0		7				
0		6				
0		5				
0		4				
0		3				
0		2				
0		1				
1	8	0	B[0,3]	B[0,2]	B[0,1]	B[0,0]

- À l'index 0 dans la mémoire cache, le bit de validité est à 1, Mais le tag vaut 4
- Il s'agit donc d'un échec car les données places dans le cache à l'index 0 ne sont pas celles correspondant à l'adresse 800
- C'est un échec, le bloc de l'@800 de la mémoire centrale est rappatrié vers la cache à l'index 0 à la place des données présentes et le tag est mis à jour.**

B[3,3] @83C	B[3,2] @838	B[3,1] @834	B[3,0] @830
B[2,3] @82C	B[2,2] @828	B[2,1] @824	B[2,0] @820
B[1,3] @81C	B[1,2] @818	B[1,1] @814	B[1,0] @810
B[0,3] @80C	B[0,2] @808	B[0,1] @804	B[0,0] @800
...	...	...	....
C[3,3] @4BC	C[3,2] @4B8	C[3,1] @4B4	C[3,0] @4B0
C[2,3] @4AC	C[2,2] @4A8	C[2,1] @4A4	C[2,0] @4A0
C[1,3] @49C	C[1,2] @498	C[1,1] @494	C[1,0] @490
C[0,3] @48C	C[0,2] @488	C[0,1] @484	C[0,0] @480
...	...	...	....
A[3,3] @43C	A[3,2] @438	A[3,1] @434	A[3,0] @430
A[2,3] @42C	A[2,2] @428	A[2,1] @424	A[2,0] @420
A[1,3] @41C	A[1,2] @418	A[1,1] @414	A[1,0] @410
A[0,3] @40C	A[0,2] @408	A[0,1] @404	A[0,0] @400