



(7.8)

همانطور که در سوال گفته شده، تعداد بیت‌های opcode برابر ۶ است. برای اینکه mapping طوری انجام شود که ۸ زیردستور العمل بتوانیم داشته باشیم، باید در سمت راست بیت‌های map شده از opcode، سه تا بیت باشد. همچنین چون حافظه کنترل دارای ۲۰۴۸ کلمه است، پس ۱۱ بیتی خواهد بود. لذا:

0 0 | X X X X X X | 0 0 0

6 bit \Rightarrow opcode

6 bit \Rightarrow mapping bits

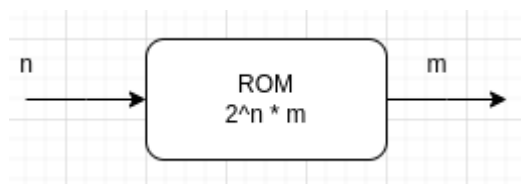
3 bit \Rightarrow تعداد بیت‌های مورد نیاز برای ذخیره ۸ زیردستور العمل برای هر روال

2 bit \Rightarrow تعداد بیت‌های کمکی در صورت نیاز به حافظه بیشتر

(چرا ۲ تا؟ چون میدونیم برای ذخیره ادرس حافظه کنترل، به ۱۱ بیت نیاز داریم:)

(7.9)

با استفاده از یک حافظه تنها خواندنی (Read Only Memory) می‌توانیم این نگاشت را انجام دهیم. به این صورت که در حافظه ROM کافی است که در آدرس n ام (تعداد بیت‌های opcode) آن، آدرس ریزدستور العمل مربوط به آن opcode ذخیره شود و کارکردی مانند شکل زیر داشته باشد:



مزیت آن می‌تواند این باشد که آدرس ریزدستور بدست‌آمده دیگر وابسته به opcode مورد استفاده نخواهد بود و می‌تواند حافظه محدودی برای تعداد زیردستورالعمل‌ها نداشته باشد و داینامیک عمل کند.

(7.13)

در صورتی که بیت Indirect برابر ۰ بود، توسط همان دستور اول، داده‌ها خوانده شده، و توسط دستور دوم، با AC جمع می‌شود.

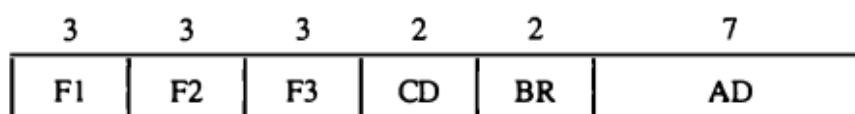
در صورتی که بیت Indirect برابر ۱ باشد، لازم است تا ابتدا ادرس موثر، درون DR ریخته شود، سپس داده را از آدرس خوانده و با AC جمع می‌شود.

پس لازم است تا در قسمت INDR2، دستورات زیر نوشته شود:

INDR2:	DRTAR	U	JMP	NEXT
	READ	U	RET	

(7.15)

با توجه به ساختار دستور برای ۲۰ بیت می‌توانیم دستورات زیر را اسخراج کنیم:



F1, F2, F3: Microoperation fields

CD: Condition for branching

BR: Branch field

AD: Address field

Figure 7-6 Microinstruction code format (20 bits).

60	:	CLRAC, COM	U	JMP	INDRCTS
61	:	WRITE, READ	I	CALL	FETCH
62	:	ADD, SUB	S	RET	63 (NETX)
63	:	DRTAC, INCDR	Z	MAP	60

مشکلهای هر خط :

۶۰ : نمیتوان همزمان هم AC را پاک کرد و هم مکمل آن را محاسبه کرد. همچنین پس از JMP به INDRCTS، دیگر به ادرس ۶۱ بر نخواهد گشت.

۶۱ : نمیتوان همزمان هم از حافظه خواند و هم بر حافظه نوشت.

۶۲ : نمیتوان همزمان عملیات جمع و تفریق را انجام داد.

۶۳ : نمیتوان همزمان یک مقدار به DR اضافه کرد و هم از آن خواند.

(7.16)

ADM:

ORG 20

NOP	I	CALL	INDRCT
READ	U	JMP	NEXT
DRTAC, ACTDR	U	JMP	NEXT
ADD	U	JMP	EXCHANGE + 2

BPNZ:

ORG 40

NOP	S	JMP	FETCH
NOP	Z	JMP	FETCH
NOP	I	CALL	INDRCT
ARTPC	U	JMP	FETCH

(7.17

ISZ:

NOP	I	CALL	INDRCT
READ	U	JMP	NEXT
INCDR	U	JMP	NEXT
DRTAC, ACTDR	U	JMP	NEXT
DRTAC, ACTDR	Z	JMP	ZERO
WRITE	U	JMP	FETCH
ZERO: WRITE, INCPC	U	JMP	FETCH

(8.2)

تاخیر کل به این صورت است:

$$30 + 80 + 10 = 120 \text{ ns}$$

در نظر داشته باشید که تاخیر موجود در دیکدر، با تاخیر MUX، همپوشانی دارد.

(8.7)

Postfix or Reverse Polish Notation

a)

$$A * B + C * D + E * F \Rightarrow A B * C D * E F * + +$$

b)

$$A * B + A * (B * D + C * E) \Rightarrow A B * A B D * C E * + * +$$

c)

$$A + B * [C * D + E * (F + G)] \Rightarrow A B C D * E F G + * + * +$$

d)

$$[A * [B + C * (D + E)]] / [F * (G + H)] \Rightarrow A B C D E + * + * F G H + * /$$

(8.13)

حافظه مورد نیاز برای آدرس‌دهی به حافظه : چون ظرفیت حافظه برابر $256k = 2^8 * 2^{10} =$

2^{18} کلمه است، پس برای اینکه بتوانیم به هرکدام از کلمه‌های حافظه اشاره کنیم، به ۱۸ بیت نیاز

داریم.

حافظه مورد نیاز برای آدرس‌دهی رجیستر : چون 60 ثبات پردازش‌گر داریم که نیاز است بتوانیم آدرس آنها را ذخیره کنیم، پس حداقل 2^6 تا آدرس باید بتوانیم نشان دهیم که برای این کار 6 بیت مورد نیاز است.

حافظه مورد نیاز برای مشخص کردن نوع آدرس‌دهی: چون 7 روش برای آدرس دادن داریم، پس حداقل به 3 بیت نیاز داریم تا بتوانیم آن را نشان دهیم.

و درد آخر چون هر دستور، در یک کلمه حافظه ذخیره می‌شود و هر کلمه حافظه هم 32 بیت است، بیت‌های مانده یعنی 5 بیت هم مربوط به opcode می‌باشد.

(8.15)

الف) آدرس نسبی، یعنی آدرس مقصد را نسبت به آدرس مبدا مشخص کنیم:

$$500 - 751 = -251$$

در آدرس دهی نسبی، دستور واقع در آدرس 750 ، در طول فاز برداشت، از مکان 750 خوانده شده و شمارنده سپس به 751 افزایش می‌یابد. پس باید مقدار آدرس نسبی را -251 قرار دهیم.

ب) چون عدد بدست آمده منفی است و همچنین این عدد مستقیماً با آدرس دستور جمع می‌شود برای اینکه این جمع هم برای اعداد منفی و هم مثبت درست باشد، پس لازم است صورت مکمل دو نمایش می‌دهیم:

$$251 = 0000\ 1111\ 1011 \Rightarrow -251 = 1111\ 0000\ 0101$$

ج)

$$PC = 751 = 0010\ 1110\ 1111$$

$$500 = 0001\ 1111\ 0100$$

$$RA = -251 = 1111\ 0000\ 0101$$

$$EA = 500 = 0001\ 1111\ 0100$$

(8.18)

در رابطه با این سوال، مثال صفحه ۲۶۴ کتاب، بخش Numerical Example را مطالعه کنید.

- a) Direct : 400
- b) Immediate : 301
- c) Relative : $302 + 400$
- d) Reg. Indirect: 200
- e) Indexed: $200 + 400 = 600$