

گزارش کار تمرین ۶ آزمایشگاه معماری

یاسمین مدنی-زهره مومنی نژاد



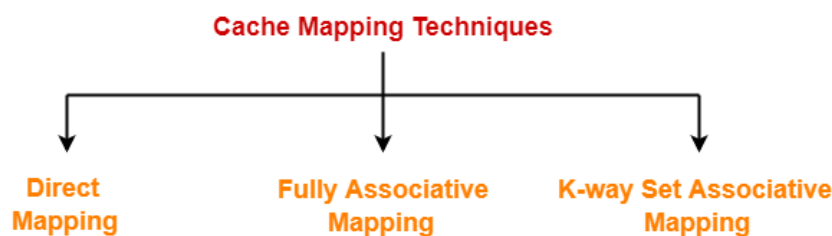
حافظه کش یا (Cache memory) که به عنوان حافظه پردازنده (CPU Memory) شناخته می شود، یک حافظه دسترسی تصادفی ایستا با سرعت بالا است که پردازنده رایانه می تواند با سرعت بیشتری نسبت به حافظه های معمولی RAM به آن دسترسی داشته باشد.

هدف از استفاده حافظه کش آن است که دستورالعمل های برنامه و داده هایی که دائماً از آن ها در عملیات های پردازشی پردازنده استفاده شود، در این حافظه قرار گیرد. پردازنده های رایانه می تواند با سرعت بسیار سریعتر از حافظه هایی نظیر RAM به اطلاعات حافظه های کش دسترسی داشته باشد و در نتیجه به خاطر افزایش سرعت دسترسی به دستورالعمل هایی که در این بخش قرار گرفته اند، سرعت کلی پردازنده افزایش پیدا می کند.

در زمانی که پردازنده رایانه در حال پردازش اطلاعات است، ابتدا به سراغ حافظه کش می رود. اگر در آنجا دستورالعمل ها و داده های مورد نیاز وجود داشته باشند، آنها را می خواند، و در عین این صورت با صرف زمان بیشتر به سراغ حافظه های بزرگتر بر روی رایانه و یا سایر تجهیزات که به آن متصل باشند می رود.

نگاشت کش تکنیکی است که به وسیله آن محتویات حافظه اصلی به حافظه کش آورده می شود.

این عمل به سه روش زیر انجام میگیرد.



نگاشت مستقیم

در حافظه کاش نیازمند آن است که هر بلوک از کش الزاما برای یک آدرس و مکان مشخص در کش اختصاص یابد. به صورت مفهومی تر، نگاشت مستقیم در حافظه کش شبیه به ردیف های یک جدول با سه ستون کار می کنند. این ستون ها شامل موارد زیر اند:

- بلوک داده (Data Block) یا خط کش (Cache line) ، که حاوی حافظه های واقعی هستند که به درون پردازنده واکشی شده و یا در حافظه کش ذخیره شده اند
- یک تگ (Tag) برای تمام یا بخشی از آدرس ها از یک داده، که اکنون در پردازنده حافظه در حال پردازش است و آدرس آن را برای واکشی را نشان میدهد؛
- یک بیت پرچم (flag bit) که نشان دهنده معتبر بودن ردیف از بیت های داده در حال پردازش و حاضر در پردازنده؛

Fully Associative

در Fully Associative، هیچ گونه ارتباط سختی بین تعداد بلوک های حافظه اصلی و خطوط حافظ کش وجود نداشته و کنترلر کش می تواند هر آدرسی را به صورت کاملاً آزادانه بارگذاری نماید، بنابراین مشکلات مرتبط با پیکربندی پیشین رفع شده و راهکاری انعطاف پذیر با آزادی عمل بالا در اختیار سیستم قرار گرفته است، به همین دلیل پیکربندی مورد بحث پر راندمان ترین در میان دو روش دیگر به شمار رفته و افزایش نرخ Hit پردازنده را به دنبال دارد.

Set-associative cache

مبادله ای بین حافظه نهان با نقشه مستقیم و حافظه نهان کاملاً انجمنی است.

کدهای مربوطه ضمیمه شده اند. در تصاویر زیر نیز عکس کد، تست بنچ و نمودار خروجی قابل مشاهده است.


```

25 library IEEE;
26 use IEEE.STD_LOGIC_1164.ALL;
27
28 -- Uncomment the following library declaration if using
29 -- arithmetic functions with Signed or Unsigned values
30 --use IEEE.NUMERIC_STD.ALL;
31
32 -- Uncomment the following library declaration if instantiating
33 -- any Xilinx primitives in this code.
34 --library UNISIM;
35 --use UNISIM.VComponents.all;
36
37 entity q6 is
38 end q6;
39
40 architecture Behavioral of q6 is
41
42     COMPONENT cache
43     PORT(
44         adrs : IN  std_logic_vector(6 downto 0);
45         dt : OUT  std_logic_vector(31 downto 0);
46         ht : OUT  std_logic;
47     );
48     END COMPONENT;
49
50
51     signal adrs : std_logic_vector(6 downto 0) := (others => '0');
52
53
54     signal dt : std_logic_vector(31 downto 0);
55     signal ht : std_logic;
56
57

```

```

54     signal dt : std_logic_vector(31 downto 0);
55     signal ht : std_logic;
56
57 BEGIN
58
59     uut: cache PORT MAP (
60         adrs => adrs,
61         dt => dt,
62         ht => ht
63     );
64
65     stim_proc: process
66     begin
67         adrs <= "0000000";
68         wait for 10 ns;
69
70         adrs <= "0110100";
71         wait for 10 ns;
72
73         adrs <= "1110100";
74         wait for 10 ns;
75
76         adrs <= "0101000";
77         wait for 10 ns;
78
79         adrs <= "0001100";
80         wait for 10 ns;
81
82         wait;
83     end process;
84
85 END;
86
87

```

Windows taskbar with search bar and application icons.

