**سوال 1.**

* واحد پردازش مرکزی (CPU) – حافظه (Memory) – درگاه های ورودی / خروجی (I/O Ports)
* هنگام انتقال اطلاعات بین واحد های درونی میکروکنترلر (CPU و Memory و I/O Ports)، سیگنال های عملیاتی (Read و Write (ارتباط با حافظه) و Chip Select(دیکدر تشخیص آدرس)) برای مشخص کردن اعمال مختلف نیاز دارند که وظیفه مهیا کردن آنها بر Control Bus است.
* حافظه یکی از واحد های اصلی میکروکنترلر هاست که وظیفه ذخیره اطلاعات را دارد. این اطلاعات شامل data و instruction است که این اطلاعات را از طریق گذرگاه داده در اختیار CPU می گذارد. حافظه به دو نوع کلی Serial Access و Random Access تقسیم می شود که هر کدام انواع مختلفی دارند.
* برای نوشتن اطلاعات 3 مرحله وجود دارد که با جابجا کردن هر کدام از این مراحل با یکدیگر ممکن است عملیات نوشتن به صورت صحیح انجام نشود. به عنوان مثال جابجایی مرحله 2 و 3 باعث ثبت اطلاعات اشتباه و آوردن مرحله 3 به ابتدا باعت ثبت اطلاعات اشتباه در مکان اشتباه خواهد شد.

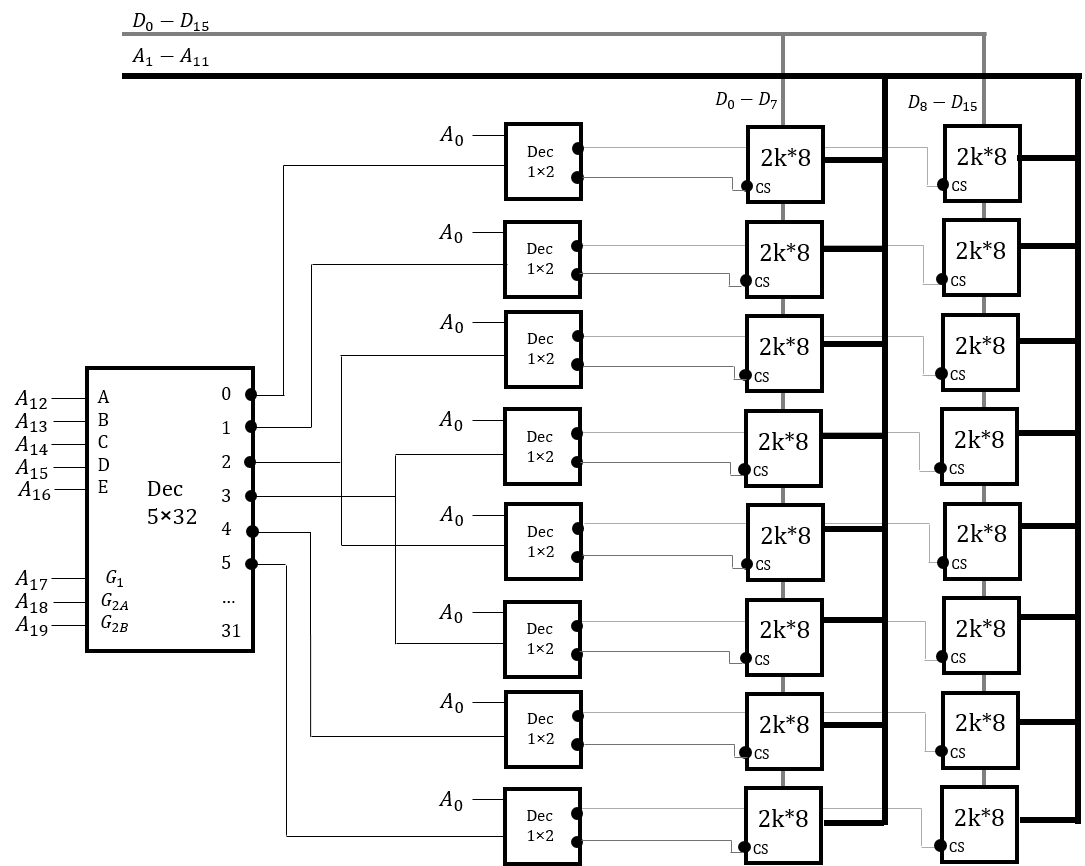
**سوال 2.**

با توجه به 16 بیتی بودن گذرگاه داده و 8 بیتی بودن عرض هر تراشه، به بانک نیاز داریم که 1 بیت کم ارزش را اشغال می کند.

از آنجایی که فضای کل حافظه RAM دارای 2 قسمت 16 کیلوبایتی است که قسمتی از آنها باهم همپوشانی دارند، پس کل فضای در دسترس باید 32 کیلوبایت باشد ولی فضای قابل استفاده باید 24 کیلوبایت باشد. بنابراین به *ردیف بانک داده نیاز داریم ولی قسمتی از آنها باهم همپوشانی منطقی دارند. (یعنی هنگام فعال شدن، 2 ردیف از قسمت همپوشانی همزمان باهم فعال می شوند و در قسمت* Chip Select *دیکودر های انتخاب* Row *این مورد لحاظ می شود.)*

با توجه به اینکه تراشه ها هستند، پس به 11 بیت برای آدرس دهی درون تراشه ای نیاز داریم. (بیت ها به تعداد گذرگاه آدرس یا 20 تاست.) (Bank Decoder و In Chip) ()

از 8 بیت باقی مانده، 5 بیت برای تشخیص دیکورد ROW ها و 3 بیت باقی مانده برای enable به کار می روند. (Bank Decoder و In Chip و Row Decoder و Enable) ()



**سوال 3.**

الف)

ب)

با توجه به اینکه تعداد segment ها 8 تاست، پس برای آدرس دهی شماره chip ها از 3 بیت با ارزش آدرس استفاده می کنیم. (Segment و In Chip) ()

از 29 بیت باقی برای آدرس دهی داخل تراشه استفاده می کنیم. (\*)

برای آدرس دهی زیربخش ها، با توجه به اندازه هر زیربخش زیربخش داریم که برای آدرس دهی آن 5 بیت نیاز داریم. (Segment و Sub Segment و In Chip) ()

از 24 بیت باقی برای آدرس دهی داخل تراشه استفاده می کنیم. (\*)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | S0 | 0000 0000 |
| S6-0 | C000 0000 |  |
|  | S1 | 2000 0000 |
| … |  |  |
|  | S2 | 4000 0000 |
| S6-3 | C300 0000 |  |
| C3FF FFFF | S3 | 6000 0000 |
| … |  |  |
|  | S4 | 8000 0000 |
| S6-6 | D600 0000 |  |
| D6FF FFFF | S5 | A000 0000 |
| ... |  |  |
|  | S6 | C000 0000 |
| S6-31 | DF00 0000 |  |
| DFFF FFFF | S7 | E000 0000 |
|  |  | FFFF FFFF |

ج)

برای 512 کیلو بایت فضای زیر زیر بخش داریم: (\*)

ابتدای زیر زیر بخش System همان ابتدای زیر بخش S6-3 و ابتدای زیر زیر بخش Flash همان ابتدای زیر بخش S6-6 است.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | S6-0 | C000 0000 |
| System | C300 0000 |  |
| C301 FFFF | … |  |
|  |  |  |
| C3FF FFFF | S6-3 | C300 0000 |
|  |  | C3FF FFFF |
|  |  | … |  |
| Flash | D600 0000 |  |
| D601 FFFF | S6-6 | D600 0000 |
|  |  | D6FF FFFF |
| D6FF FFFF | … |  |
|  |  |  |
|  |  |  | S6-31 | DF00 0000 |
|  |  |  | DFFF FFFF |

\* برای رسیدن به آدرس شروع هر خانه ، این مقدار را با آدرس شروع خانه قبلی جمع می کنیم. برای رسیدن به آدرس انتهای هر خانه، این مقدار را با آدرس شروع خانه جمع و یکی از آن کم می کنیم. (از این قانون در بالا استفاده شده است.)