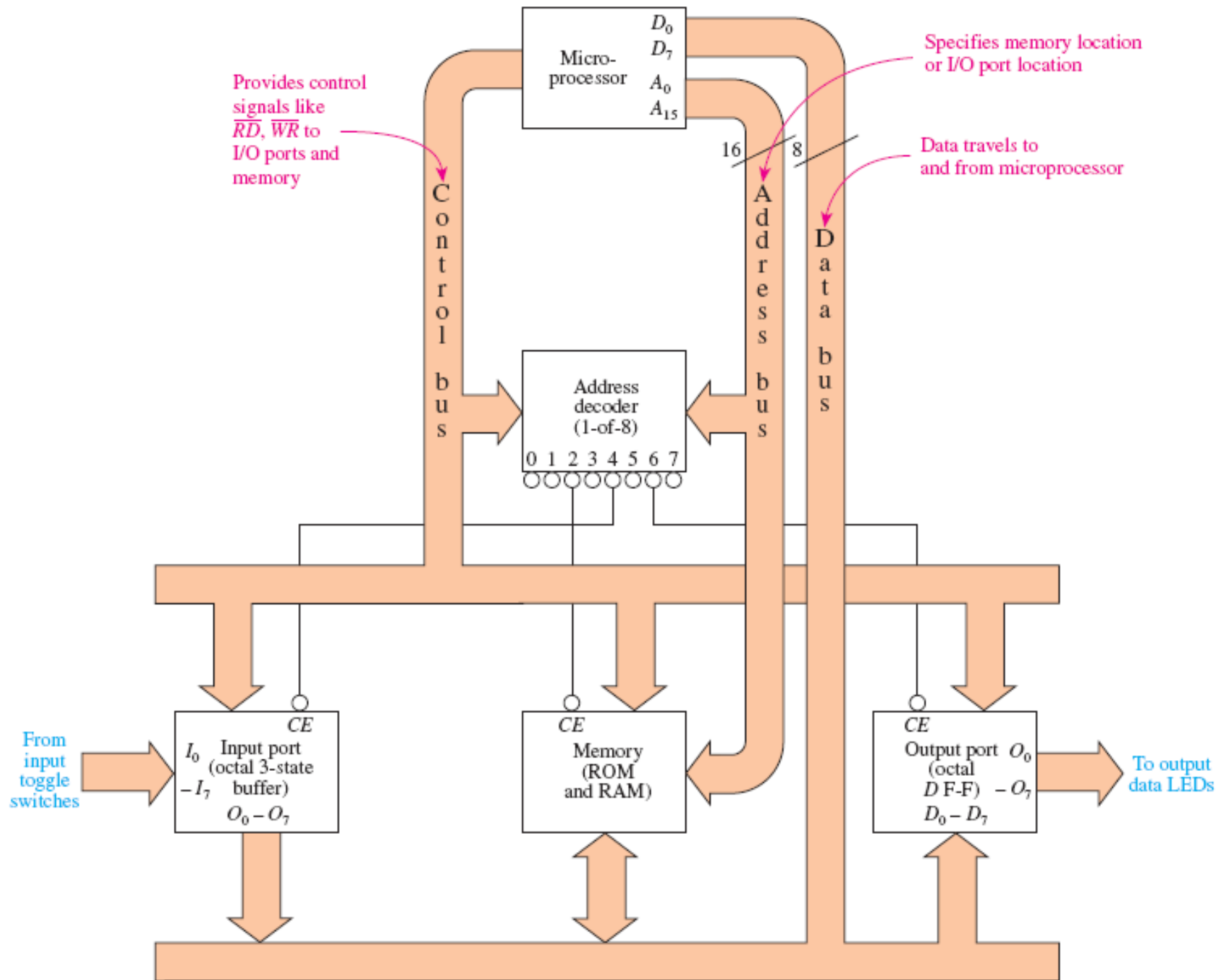


Mikroprozessor



Mikroprocesor i komunikacione linije

- ↙ Mikroprocesor
 - Izvršavanje aritmetičko logičkih operacija, akumulator
- ↙ Adresna linija (address bus):
 - ↖ specifikacija adrese
- ↙ Kontrolna linija (control bus):
 - ↖ naredna komanda
- ↙ Linija podataka (data bus):
 - ↖ bidirekcionni prenos iz procesora u memoriju i obrnuto

Ostale komponente

↙ Dekoder adrese

↖ Osigurava da samo jedna lokacija u memoriji bude aktivna (za čitanje ili pisanje od strane linije podataka)

↙ Memorija

↖ ROM – pri pokretanju računara (aktiviranje perifernih uređaja, pokretanje OS-a)

↖ RAM – privremena memorija (promenljiva), korisnički programi itd.

↙ Ulazni port

↖ Veza sa nekim ulaznim uređajem: miš, tastatura...

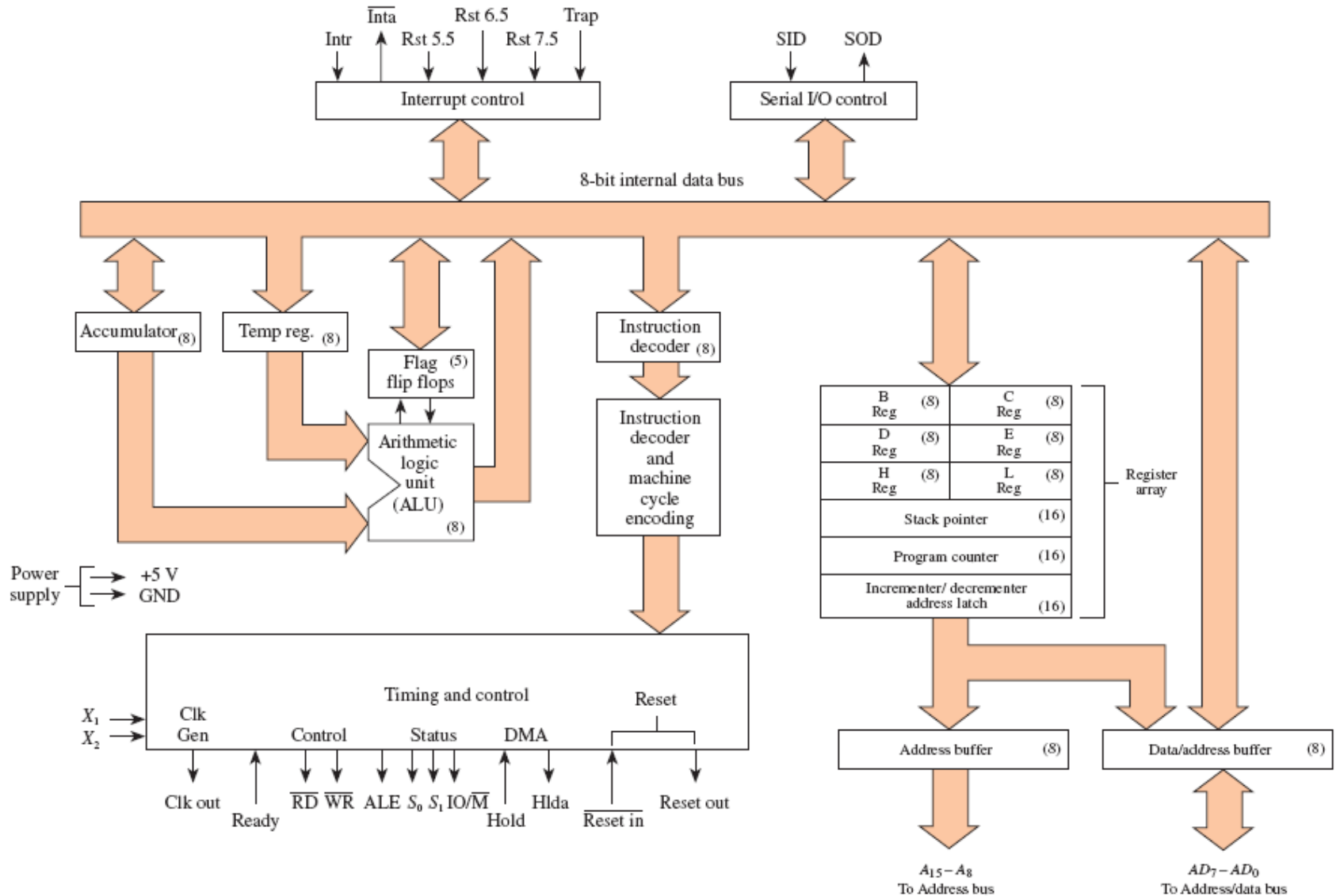
↙ Izlazni port

↖ Veza sa izlaznim uređajem: štampač, monitor...

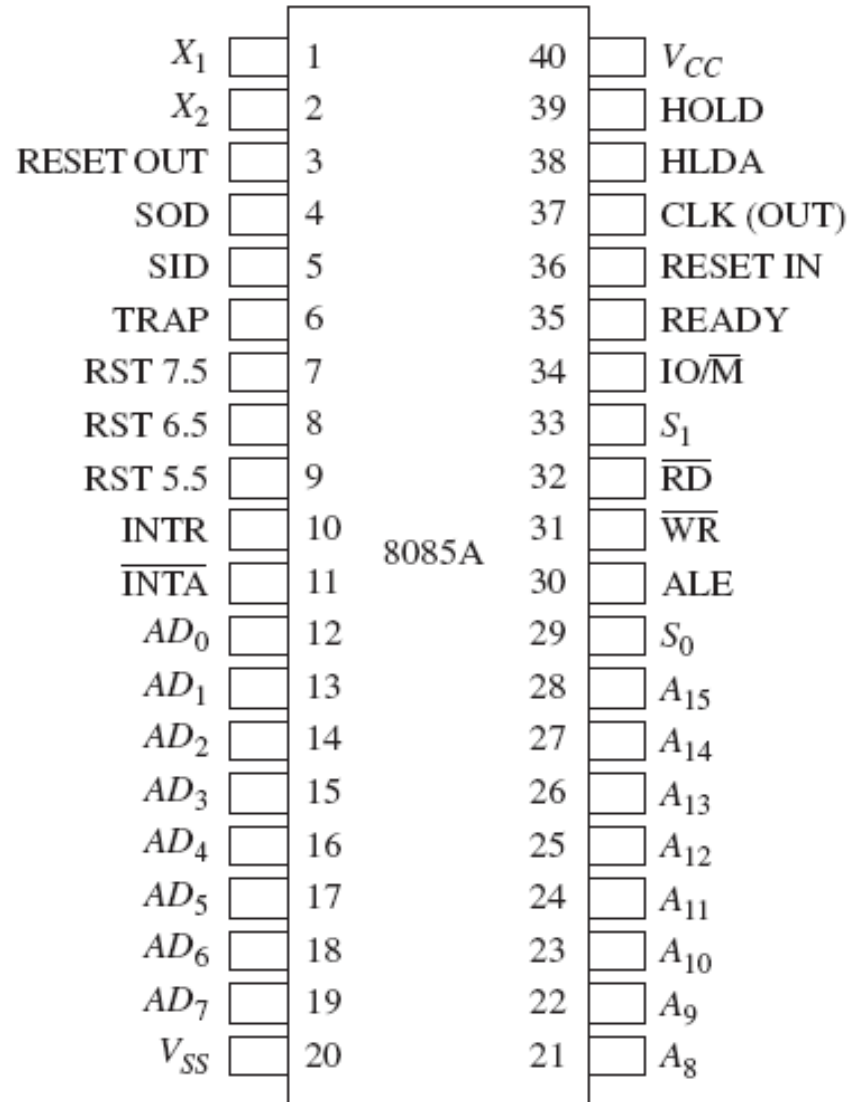
Softverski aspekti mikroprocesora

- Akumulator (8 bita)
- Asemblerski jezik (assembly language)
 - LDA addr – učitava iz memorije u akumulator
 - STA addr – iz akumulatora u memoriju
- Mikroprocesor se brine za tajming između tri komunikacione linije

Funkcionalna organizacija Intel 8085A



Intel 8085A – konfiguracija portova



Organizacija komponenti

- Akumulator A (8 bita) povezan na liniju podataka
- Još dodatnih 6 8-bitnih registara opšte namene: B, C, D, E, H i L takođe povezani na liniju podataka
- ALU jedinica
 - radi sve aritmetičko-logičke operacije u kombinaciji sa akumulatorom i pomoćnim registrom
 - Rezultati ALU operacija se šalju na liniju podataka i na 5 fleg flip-flova

Organizacija komponenti - nastavak

- Instrukcijski registar i dekodler:
 - prihvataju softverske instrukcije iz memorije i kreiraju tajming i kontrolne signale potrebne za izvršavanje instrukcija
- Kontrola prekida
 - Obustavljanje sekvencijalnog izvršavanja instrukcija
- Tri dodatna 16-bitna registra:
 - SP – pokazivač na vrh steka
 - PC – adresa sledeće instrukcije
 - AL – trenutna adresa koja se šalje na adresnu liniju

Primer izvršavanja programa

- Program:

LDA 4000H

STA 6000H

- Pretpostavimo da program počinje na adresi 2000H, tj. da je PC=2000H inicijalno
- Program je u memoriji, naravno, zapisan binarnim (mašinskim kodom)

Primer izvršavanja programa

1. PC postavlja adresu 2000H na adresnu liniju
2. Kontrolna jedinica šalje signal RD za uzimanje podataka na kontrolnu liniju, što uzrokuje da se sadržaj sa lokacije 2000H stavi na liniju podataka (vrednost je 3AH)
3. 3AH stiže u instrukcijski registar i dekodeer shvata da je ovo LDA instrukcija, za čim bi trebalo da usledi 2 bajtna adresa. PC se inkrementuje 2 puta, a u LA se upisuje dvobajtna adresa

Primer izvršavanja programa

4. Adresna linija i LA sada imaju vrednost 4000H
5. Kontrolna jedinica šalje RD signal ponovo, i sadržaj sa lokacije 4000H se upisuje na liniju podataka
6. Potom taj sadržaj sa linije podataka stiže u akumulator

Primer izvršavanja programa

Sada deo sa operacijom STA 6000H

1. Nakon izvršavanja LDA 4000H, PC=2003H.
2. Kontrolna jedinica šalje RD signal, što ima za efekat uzimanje podataka sa lokacije 2003H i smeštanje na liniju podataka. Sadržaj lokacije je 32H.
3. Dekoder shvata da je u pitanju STA instrukcija i zna da mora da uslede još dve bajtne lokacije za adresu pa povećava PC dva puta i učitava 16-bitnu adresu u LA

Primer izvršavanja programa

4. Adresna linija sada ima sadržaj 6000H

5. Instrukcijski dekodler sada izaziva kontrolni signal smeštanja sadržaja akumulatora na liniju podataka.

6. Šalje se kontrolni signal WR koji ima za efekat da se ono zapisano na liniji podataka upiše u memoriji na lokaciji 6000H

Prevođenje asemblerskog jezika u mašinski kod

- Veoma jednostavno, samo se koristi tabela preslikavanja
- Kod viših programskih jezika je potreban kompajler! Ovde nešto nalik na search and replace.
- Asemblerski jezik omogućava da se isti programi pišu za različite mikroprocesore. Samo se posle iskoristi druga tabela preslikavanja.